Final Technical Report to ETRI

Enhancing Performance of Multi-Node Mesh Networks Using Signal Processing Techniques

PI: Xiaoli Ma

Team: Dr. Sungeun Lee, Giwan Choi, Hayang Kim, Malik Muhammad Usman Gul

Acknowledgment: ETRI

January 31, 2011

Contents

List of Figures

List of Tables

Chapter 1

Summary

1.1 Deliverables

- Publications
  1. "An Improved Synchronization Scheme for OFDMA Systems with Initial Ranging Transmissions," *presented on ASILOMAR 2010*
  2. "Null sub-carrier based carrier frequency offset estimation for OFDMA," *presented on IEEE DSP/SPE Workshop 2010*
  3. "Timing adjustment techniques to mitigate interference between multiple nodes in OFDMA mesh networks," *accepted to ICASSP 2011*

  4. "Tracking Low-Precision Clocks with Time-Varying Drifts Using Kalman Filtering,", *submitted on IEEE Trans. on Networking*

- Patents
  1. ID-5479: "HR technique for ranging process," *disclosure is submitted to GTRC OTL*
  2. ID-5485: "CFO estimation and compensation method for OFDMA," *disclosure is submitted to GTRC OTL*

Chapter 2

An Improved Synchronization Scheme for OFDMA Systems with Initial Ranging Transmissions

At the initial ranging process, it is crucial to estimate multi-user's multiple parameters such as ranging code, timing and frequency offset and to identify these multiple parameters for each user. This chapter presents an improved parameter estimation and pairing algorithm for initial ranging process in OFDMA uplink systems by exploiting multidimensional harmonic retrieval (HR) technique. Unlike most existing techniques that estimate each parameters independently and associate the ranging codes with the estimated parameters manually, the proposed method estimates multiple user's ranging code, timing and frequency offset simultaneously, and pair up all the multiple parameters automatically. The simulation results confirm that the proposed technique not only improves the ranging code detection capability and adjusts the acquisition range of the estimators, but also increases the maximum number of resolvable users for a given samples.

2.1 Introduction

In general OFDMA uplink systems, to maintain the orthogonality among multiple users, the signals from all active users should arrive at the base station (BS) synchronously [3] [4].

Recently, some initial ranging algorithms are introduced to detect multi-user's ranging code as well as to estimate the timing and frequency offsets [4] [5] [6]. However, the most important task for initial ranging process beyond the accurate code detection and offset estimation is the *pairing* between the detected codes and the estimated offsets. It is mainly because multiple RSSs can be only distinguished by the group of ranging codes at the BS which each RSS transmitted. Since the BS should give a feedback (containing the successful code detection, timing and frequency offset adjustment) to each RSS, the parameter pairing should be required.

In order to bind the detected ranging codes with the estimated timing or frequency offsets, almost existing works either sacrifices the estimation performance by inserting the code sequence in the estimated parameters [6] or performs exhaustive full search by correlating all sets of ranging codes and offsets at the BS [4] [5].

In this chapter, we propose an improved ranging technique using harmonic retrieval (HR) used for multidimensional frequency estimation. By exploiting HR technique for initial ranging process, all the detected ranging codes, the estimated timing offsets and frequency offsets of multiple users can be automatically paired, and no extra ranging code and offset association process is required. In addition, the HR technique enables the BS to increase the maximum number of resolvable users the BS can distinguish.

2.2 System Model

Let us consider an OFDMA uplink system employing N subcarriers. Among the whole subcarriers, virtual subcarriers are placed at both edges of the spectrum to prevent the spectrum aliasing. Except the virtual subcarriers, the useful subcarriers are grouped by multiple subchannels, and these subchannels are assigned to multiple users (subscribers) for transmission. Typically, each subchannel is divided into Q subbands composed of a set of V adjacent subcarriers [7] [5] [6]. When one transmission block consists of M consecutive OFDMA symbols, let us define a bunch of consecutive V subcarriers (1 subband) and M OFDMA symbols as a *tile*. The overall ranging structure is described in Figure 2.1

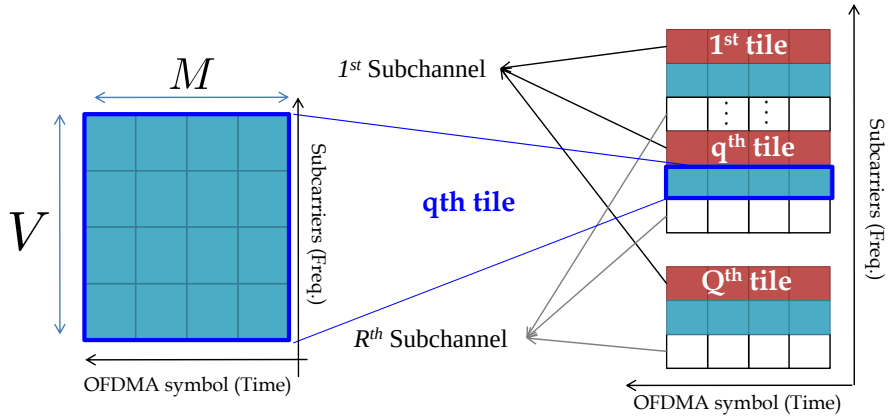


Figure 2.1: The *tile* structure for OFDMA systems

Now, among multiple subchannels, let us define RSSs only use R subchannels for ranging with M consecutive OFDMA symbols, and then the total number of subcarriers used for ranging is $N_R = RQV$.

After the fast Fourier transform (FFT) at the receiver, the received data at the q-th tile of the ranging subchannel is written as

$$[\mathbf{X}_q]_{v,m} = \sum_{k=0}^{K-1} [\mathbf{C}_k]_{v,m} H_k[i_q + v] e^{\frac{j2\pi}{N}(i_q+v)\theta_k} e^{\frac{j2\pi N_T}{N} m\epsilon_k} + w_m[i_q + v], \quad \forall q \in [0, Q), \forall v \in [0, V), \forall m \in [0, M), \tag{2.1}$$

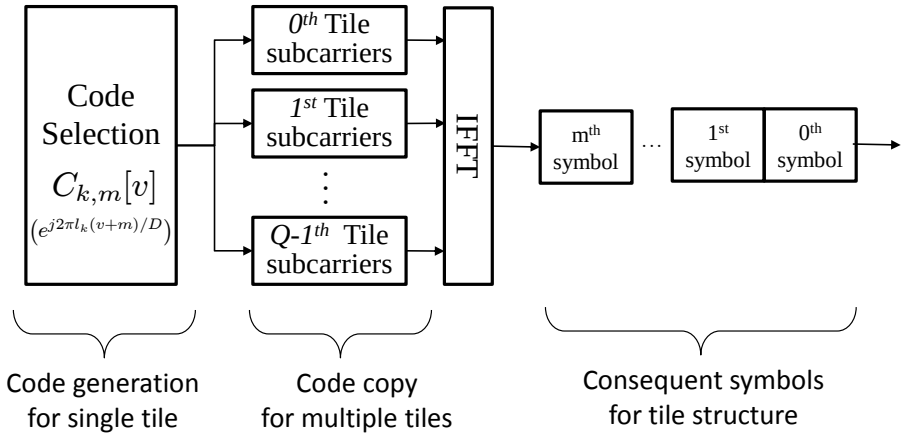


Figure 2.2: The transmitter block diagram with code

where $[\mathbf{x}]_{v,m}$ stands the v-th subcarrier of the m-th OFDMA symbol in the tile, $N_T = N + N_G$ with N_G guard interval for ranging symbols, i_q is the starting subcarrier index for the q-th tile, θ_k and ϵ_k denote the timing and frequency offset of the k-th RSS, respectively, and $w_m[v]$ is an additive white Gaussian noise (AWGN) with zero mean and variance σ_w^2. $\mathbf{C}_k$ is the code sequence matrix for the k-th RSS.

2.2.1 Code Design

The code sequence for the k-th RSS user is defined as

$$[\mathbf{C}_k]_{v,m} = e^{j2\pi v \frac{t_k}{C_T}} e^{j2\pi m \frac{f_k}{C_F}}, \tag{2.2}$$

$$[\mathbf{C}_k]_{v,m} = e^{j2\pi v \cdot t_k/C_T} e^{j2\pi m \cdot f_k/C_F}, \tag{2.3}$$

where $\mathcal{I}_k = (t_k, f_k)$ is the code index pair for the k-th user which is chosen from the whole code index set $\mathcal{I}$, i.e., $\mathcal{I}_k \in \mathcal{I}$. In addition, C_T and C_F are the design parameters to determine the number of RSSs supporting at the time and frequency domain, respectively. Note that C_T and C_F can be adjusted according to RSS's timing and frequency distributions.

Since the subband size in one tile is smaller than the channel coherence bandwidth [5] [6], the received output can be expressed as an average channel response as follows:

$$[\mathbf{X}_q]_{v,m} = \sum_{k=0}^{K-1} [\mathbf{h}_q]_k \, e^{j2\pi v \eta_k} e^{j2\pi m \xi_k} + [\mathbf{W}_q]_{v,m} \tag{2.4}$$

where $[\mathbf{h}_q]_k$ is the phase rotated average channel response for the k-th user at the q-th tile written as

$$[\mathbf{h}_q]_k = \sum_{v=0}^{V-1} H_k[i_q + v] e^{j2\pi i_q \theta_k/N}.$$

Compared with (2.1), the code sequence of the k-th user, $[\mathbf{C}_k]_{v,m}$, is absorbed into the *effective* timing and frequency offsets η_k and ξ_k, and this *effective* timing and frequency offsets of the k-th user are given by

$$\eta_k = \frac{t_k}{C_T} - \frac{\theta_k}{N}, \tag{2.5}$$

$$\xi_k = \frac{f_k}{C_F} + \frac{\epsilon_k N_T}{N}. \tag{2.6}$$

2.3 Multidimensional Folding (MDF) Estimation using Multiple Tiles

In order to exploit automatic pairing property of harmonic retrieval (HR), we formulate the received signal as a 2-D mixture form, and then apply MDF estimation method [8] for estimating timing and frequency offset of RSSs as well as detecting RSSs codes simultaneously.

2.3.1 2-D mixture model

Let us recall the received symbols at the q-th tile is $\mathbf{X}_q \in \mathbb{C}^{V\times M}$. Given (2.4), $\mathbf{X}_q$ can be written as 2-D mixture matrix form as

$$\mathbf{X}_q = \mathbf{T}\mathbf{D}\left(\mathbf{h}_q\right)\mathbf{F}^T + \mathbf{W}_q, \tag{2.7}$$

where $\mathbf{T} \in \mathbb{C}^{V\times K}$ and $\mathbf{F} \in \mathbb{C}^{M\times K}$ are the transpose of Vandermonde matrices with the common ratios $e^{j2\pi v\eta_k}$ and $e^{j2\pi m\xi_k}$ for each column to represent the timing and frequency offsets of K RSSs, respectively. $\mathbf{D}\left(\mathbf{h}_q\right) \in \mathbb{C}^{K\times K}$ is the diagonal matrix with the k-th entry $[\mathbf{h}_q]_k$. For simplicity, let us define the (v,m)-th element of the matrix $\mathbf{X}_q$ as $x_{q,v,m}$. Then, given (2.7), we can define the sample vector form $\mathbf{x}_q$, which is

$$\begin{aligned}\mathbf{x}_q &= \begin{bmatrix} x_{q,1,1} & x_{q,1,2} & \cdots & x_{q,1,M} & x_{q,2,1} & \cdots & x_{q,V,M}\end{bmatrix}^T \\ &= (\mathbf{T}\odot\mathbf{F})\mathbf{h}_q,\end{aligned} \tag{2.8}$$

where $\mathbf{x}_q \in \mathbb{C}^{MV\times 1}$ and $\odot$ is the Khatri-Rao product (vector-wise Kronecker product) [9]. Since the received signal is formed as undamped 2-D exponentials for each tile, we can apply multidimensional folding (MDF) algorithm [8] for each tile at the initial ranging process.

2.3.2 MDF algorithm using Multiple Tiles

Smoothing operation

As shown in (2.7) and (2.8), the observed symbols at each tile can be described as the special form of the multiple harmonic combinations. Based on this structure, we apply *smoothing operator* $\mathcal{S}(\mathbf{x})$ in order to fully utilize this special characteristics of the observed symbols and improve the estimation performance.

Let us define a set of 2-dimensional selection matrices $\mathbf{J}_{v_2m_2} \in \mathbb{C}^{V_1M_1 \times VM}$ given by

$$\begin{aligned} \mathbf{J}_{v_2m_2} &= \mathbf{J}_{v_2} \otimes \mathbf{J}_{m_2}, \\ \mathbf{J}_{v_2} &= \begin{bmatrix} \mathbf{0}_{V_1\times(v_2-1)} & \mathbf{I}_{V_1} & \mathbf{0}_{V_1\times(V_2-v_2)} \end{bmatrix}, \qquad v_2 = 1, \ldots, V_2, \\ \mathbf{J}_{m_2} &= \begin{bmatrix} \mathbf{0}_{M_1\times(m_2-1)} & \mathbf{I}_{M_1} & \mathbf{0}_{M_1\times(M_2-m_2)} \end{bmatrix}, \qquad m_2 = 1, \ldots, M_2, \end{aligned} \tag{2.9}$$

where V_1, V_2 and M_1, M_2 are positive integers satisfying $V_1 + V_2 = V$ and $M_1 + M_2 = M$.

Using (2.8) and (2.9), let us define *smoothing operator* $\mathcal{S}(\mathbf{x})$ and the smoothed matrix $\mathbf{X}_{q,\mathcal{S}}$ for the sample vector as

$$\begin{aligned} \mathbf{X}_{q,\mathcal{S}} &= \mathcal{S}\left(\mathbf{x}_q\right) \\ &= \begin{bmatrix} \mathbf{J}_{11}\mathbf{x}_q & \mathbf{J}_{12}\mathbf{x}_q & \cdots & \mathbf{J}_{1M_1}\mathbf{x}_q & \mathbf{J}_{21}\mathbf{x}_q & \cdots & \mathbf{J}_{V_1M_1}\mathbf{x}_q \end{bmatrix} \end{aligned} \tag{2.10}$$

where $\mathbf{X}_{q,\mathcal{S}} \in \mathbb{C}^{V_1M_1 \times V_2M_2}$. Here, we show an explicit expression of $\mathbf{X}_{q,\mathcal{S}}$ using $x_{q,v,m}$ in order to figure out the smoothing operator,

$$\mathbf{X}_{q,\mathcal{S}} = \begin{bmatrix} \mathbf{X}_{q,1} & \mathbf{X}_{q,2} & \cdots & \mathbf{X}_{q,V_2} \\ \mathbf{X}_{q,2} & \mathbf{X}_{q,3} & \cdots & \mathbf{X}_{q,V_2+1} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{X}_{q,V_1} & \mathbf{X}_{q,V_1+1} & \cdots & \mathbf{X}_{q,V} \end{bmatrix}$$

with $\mathbf{X}_{q,v} \in \mathbb{C}^{M_1 \times M_2}$

$$\mathbf{X}_{q,v} = \begin{bmatrix} x_{q,v,1} & x_{q,v,2} & \cdots & x_{q,v,M_2} \\ x_{q,v,2} & x_{q,v,3} & \cdots & x_{q,v,M_2+1} \\ \vdots & \vdots & \ddots & \vdots \\ x_{q,v,M_1} & x_{q,v,M_1+1} & \cdots & x_{q,v,M} \end{bmatrix}.$$

In the absence of noise, it is already verified in [8, Lemma 2] that (2.10) can be decomposed into harmonic matrices as follows:

$$\mathbf{X}_{q,\mathcal{S}} = \mathcal{S}\left(\mathbf{x}_q\right) = \mathbf{G}_1\mathbf{D}(\mathbf{h}_q)\mathbf{G}_2^T \tag{2.11}$$

where

$$\begin{aligned} \mathbf{G}_1 &= \mathbf{T}^{(V_1)} \odot \mathbf{F}^{(M_1)} \\ \mathbf{G}_2 &= \mathbf{T}^{(V_2)} \odot \mathbf{F}^{(M_2)} \end{aligned}$$

with $V_1 + V_2 = V + 1$, $M_1 + M_2 = M + 1$.

In order to further explore the data structure, we operate the backward smoothing as well as forward smoothing on the sample vector $\mathbf{x}_q$ in (2.8). Let us define a backward sample vector $\mathbf{y}_q = \Pi_{VM} \cdot \mathbf{x}_q^*$ where Π_{VM} is an $VM \times VM$ backward permutation matrix. By using the property of

harmonics, it is easily verified that

$$\begin{aligned}\mathbf{y}_q &= \Pi_{VM} \cdot (\mathbf{T} \odot \mathbf{F})^* \mathbf{h}_q^* \\ &= (\mathbf{T} \odot \mathbf{F}) \mathbf{D}(\mathbf{e}_\beta) \mathbf{h}_q^*, \\ &= (\mathbf{T} \odot \mathbf{F}) \tilde{\mathbf{h}}_q, \end{aligned} \tag{2.12}$$

where $\tilde{\mathbf{h}}_q = \mathbf{D}(\mathbf{e}_\beta)\mathbf{h}_q^*$ and $\mathbf{e}_\beta = \begin{bmatrix} e^{-j\beta_1} & e^{-j\beta_2} & \cdots & e^{-j\beta_K} \end{bmatrix}^T$ with $\beta_k = (V-1)\eta_k + (M-1)\xi_k$. Since (2.12) has the same harmonic matrix form as (2.8), we can use both forward and backward sample vectors to estimate the phase. In the similar way like (2.10), $\mathbf{Y}_{q,\mathcal{S}}$ can be also obtained by backward smoothing operation:

$$\mathbf{Y}_{q,\mathcal{S}} = \mathcal{S}(\mathbf{y}_q) = \mathbf{G}_1 \mathbf{D}(\tilde{\mathbf{h}}_q) \mathbf{G}_2^T. \tag{2.13}$$

Multiple tile combination

Unlike [10], after calculating the smoothed matrices $\mathbf{X}_{q,\mathcal{S}}$ and $\mathbf{Y}_{q,\mathcal{S}}$ for each tile, we collect all the smoothed matrices in order to perform MDF algorithm:

$$\tilde{\mathbf{X}} = [\mathbf{X}_{1,\mathcal{S}} \quad \mathbf{X}_{2,\mathcal{S}} \cdots \mathbf{X}_{Q,\mathcal{S}} \quad \mathbf{Y}_{Q,\mathcal{S}} \quad \mathbf{Y}_{Q-1,\mathcal{S}} \cdots \mathbf{Y}_{1,\mathcal{S}}] \tag{2.14}$$

$$= \mathbf{G}_1 (\mathbf{G}_2 \odot \mathbf{B})^T \tag{2.15}$$

$$= \mathbf{G}_1 \hat{\mathbf{G}}_2^T \tag{2.16}$$

where $\tilde{\mathbf{X}} \in \mathbb{C}^{V_1 M_1 \times 2QV_2M_2}$ and $\mathbf{B} \in \mathbb{C}^{2Q \times K}$, $\mathbf{H} \in \mathbb{C}^{K \times Q}$, $\tilde{\mathbf{H}} \in \mathbb{C}^{K \times Q}$ as follows:

$$\mathbf{B} = \begin{bmatrix} \mathbf{H}^T \\ \Pi_Q \tilde{\mathbf{H}}^T \end{bmatrix}, \quad \mathbf{H} = \begin{bmatrix} \mathbf{h}_1 \ \mathbf{h}_2 \cdots \mathbf{h}_Q \end{bmatrix} \quad \tilde{\mathbf{H}} = \begin{bmatrix} \tilde{\mathbf{h}}_1 \ \tilde{\mathbf{h}}_2 \cdots \tilde{\mathbf{h}}_Q. \end{bmatrix}$$

Eigenvector based estimation

$\tilde{\mathbf{X}}$ is of rank K almost surely. Therefore, the singular valude decomposition (SVD) of $\tilde{\mathbf{X}}$ yields

$$\tilde{\mathbf{X}} = \mathbf{U}_s \mathbf{\Sigma}_s \mathbf{V}_s^H$$

where $\mathbf{U}_s$ has K columns that together span the column space of $\tilde{\mathbf{X}}$. Therefore, there exists and $K \times K$ nonsingular matrix $\mathbf{L}^{-1}$ such that $\mathbf{U}_s = \mathbf{G}\mathbf{L}^{-1}$. Based on the IMDF algorithm in [8] and [11], we get the matrix $\mathbf{P}$ which is the Khatri-Rao product of two Vandermonde matrices, and there are K columns in $\mathbf{P}$. The phase rotation components caused by timing offset and frequency offset appear in the same column of the $\mathbf{P}$. In other words, for a fixed k-th column, $e^{j2\pi\eta_k}$ and $e^{j2\pi\xi_k}$ appear simultaneously in the same column of $\mathbf{P}$. Thanks to this combined structure, we can estimate K RSS's timing offset and frequency offset by dividing suitably chosen elements of the $\mathbf{P}$. More specifically, $\mathbf{P}$ can be expressed

as

$$\mathbf{P} = \begin{bmatrix} \alpha_{\eta_1}\alpha_{\xi_1} & \alpha_{\eta_2}\alpha_{\xi_2} & \cdots & \alpha_{\eta_K}\alpha_{\xi_K} \\ \alpha_{\eta_1}\alpha_{\xi_1}^2 & \alpha_{\eta_2}\alpha_{\xi_2}^2 & \cdots & \alpha_{\eta_K}\alpha_{\xi_K}^2 \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_{\eta_1}\alpha_{\xi_1}^{M_1} & \alpha_{\eta_2}\alpha_{\xi_2}^{M_1} & \cdots & \alpha_{\eta_K}\alpha_{\xi_K}^{M_1} \\ \alpha_{\eta_1}^2\alpha_{\xi_1} & \alpha_{\eta_2}^2\alpha_{\xi_1} & \cdots & \alpha_{\eta_K}^2\alpha_{\xi_1} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_{\eta_1}^{V_1-1}\alpha_{\xi_1}^{M_1} & \alpha_{\eta_2}^{V_1-1}\alpha_{\xi_2}^{M_1} & \cdots & \alpha_{\eta_K}^{V_1-1}\alpha_{\xi_K}^{M_1} \end{bmatrix}. \tag{2.17}$$

where $\alpha_{\eta_k} = e^{j2\pi\eta_k}$ and $\alpha_{\xi_k} = e^{j2\pi\xi_k}$

2.3.3 Effective Timing Offset and Frequency Offset Estimation

Thanks to the special structure of $\mathbf{P}$ shown in (2.17), we can estimate the *effective* timing and frequency offset of each RSS simultaneously, and automatically pair each timing offset with frequency offset.

The effective timing offset of the k-th RSS can be estimated as

$$\widehat{e^{j2\pi\eta_k}} = \frac{1}{(V_1-2)M_1} \sum_{p=1}^{(V_2-1)M_1} \frac{[\mathbf{P}]_{p+M_1,k}}{[\mathbf{P}]_{p,k}}, \qquad \forall k = 1, \ldots, K \tag{2.18}$$

Finally, the effective timing offset for the k-th RSS can be obtained by

$$\hat{\eta}_k = \frac{1}{2\pi} Im\left(\ln \widehat{e^{j2\pi\eta_k}}\right). \tag{2.19}$$

In the similar way, the effective frequency offset of the k-th RSS can be estimated as

$$\widehat{e^{j2\pi\xi_k}} = \frac{1}{(V_1-1)(M_1-1)} \sum_{\substack{p=1, \\ \bmod (p,M_1)\neq 0}}^{(V_1-1)M_1} \frac{[\mathbf{P}]_{p+1,k}}{[\mathbf{P}]_{p,k}}, \qquad \forall k = 1, \ldots, K \tag{2.20}$$

Finally, the effective frequency offset for the k-th RSS can be obtained by

$$\hat{\xi}_k = \frac{1}{2\pi} Im\left(\ln \widehat{e^{j2\pi\xi_k}}\right). \tag{2.21}$$

2.3.4 Code Detection

Recall the *effective* timing and frequency offset is composed of the code index as well as actual timing and frequency offset, as shown in (2.5) and (2.6). Since $\hat{\eta}_k$ and $\hat{\xi}_k$ are automatically paired, the paired ranging code, timing offset and frequency offset of the k-th RSS can be detected and estimated by using the effective timing and frequency offsets.

First, let us detect the code index t_k absorbed in the effective timing offset. The code index t_k is integer value, so let us define $\Delta i = \frac{\theta_{\max}}{2}\frac{C_T}{N}$. Then, the formula $C_T \cdot \eta_k + \Delta i$ belongs to the range $I_{t_k} = [t_k - \Delta i, t_k + \Delta i]$. If $\theta_{\max} < N/C_T$, it is guaranteed that there is only one integer t_k within the

range I_{t_k}. In this case, the effective timing offset can be univocally mapped into their corresponding codes and an code index estimate of t_k is found as

$$\hat{i}_{T,k} = \lfloor C_T \cdot \hat{\eta}_k + \Delta i \rfloor \tag{2.22}$$

where $\lfloor x \rfloor$ means the closest integer smaller than x.

In the similar way, we can estimate the code index f_k absorbed in the effective frequency offset. Since the code index f_k is also integer value, we can easily calculate $\hat{i}_{F,k}$ based on (2.6) and (2.21) as follows:

$$\hat{i}_{F,k} = \lfloor C_F \cdot \hat{\xi}_k \rfloor. \tag{2.23}$$

Since the code indices $\hat{i}_{T,k}$ and $\hat{i}_{F,k}$ are coupled together, we can improve the code detection performance by utilizing both code indices simultaneously. Let us define the estimate code index pair $\hat{\mathcal{I}}_k = (\hat{i}_{T,k}, \hat{i}_{F,k})$. Then, the code detection strategy is given by

$$\hat{\mathcal{I}}_k \text{ is declared as } \begin{cases} \text{detected to } \mathbf{C}_k, & \hat{\mathcal{I}}_k \neq \hat{\mathcal{I}}_l \quad \forall k \neq l, \quad \hat{\mathcal{I}}_k = \mathcal{I}_k \in \mathcal{I} \\ \text{undetected}, & \text{otherwise} \end{cases} \tag{2.24}$$

.

The first condition determines whether it is the phantom code index or not. Here, the phantom code means the duplicated code detection at the code detection procedure. The second condition investigate whether the detected code index pair is true code index pair or not.

2.3.5 Actual Timing and Frequency Estimation

After detecting and verifying the code index, finally Let us assume $\hat{\mathcal{I}}_k = (\hat{i}_{T,k}, \hat{i}_{F,k})$ is correctly detected code index pair. Then, using these code indices, we finally determine the actual timing and frequency offset for the k-th RSS.

Based on (2.5) and (2.19), the estimated actual timing offset for the k-th RSS is given by

$$\hat{\theta}_k = N \left(\frac{\hat{i}_{T,k}}{C_T} - \hat{\eta}_k \right). \tag{2.25}$$

In the same way, the estimated actual frequency offset for the k-th RSS is written by

$$\hat{\epsilon}_k = \frac{N}{N_T} \left(\hat{\xi}_k - \frac{\hat{i}_{F,k}}{C_F} \right). \tag{2.26}$$

2.4 Statistical Characteristics

2.4.1 Identifiability

When the estimator uses the timing offset dimension as a primary dimension, the maximum number of RSSs, which is almost surely uniquely identifiable, is defined as

$$K_{\max} = \min\left\{(V_1 - 1)M_1, 2QV_2M_2\right\}. \tag{2.27}$$

By adjusting the smoothing factors V_1, V_2, M_1 and M_2, the code detection and timing and frequency offset estimation performance can be improved as well as $K_{\max}$ can be increased.

2.4.2 Acquisition range

In general, the acquisition range not to make ambiguity at the code detection is summarized as

$$\theta_{\max} < \frac{N}{C_T}, \tag{2.28}$$

$$|\epsilon_{\max}| < \frac{N}{2N_TC_F}. \tag{2.29}$$

However, the acquisition range for the timing or frequency offset and the maximum number of supportable users can be changed depending on how to design the code index pair $\mathcal{I}_k$ and parameters C_T and C_F.

Usually for safety, we setup the design parameters C_T and C_F smaller than $K_{\max}$ because there is a trade-off between the maximum allowable estimation range and the number of supportable RSSs by adjusting C_T and C_F.

2.4.3 Code design

Now, let us define various code index pair scenarios and compare the identifiability and acquisition range for each scenario.

Paired code, $\mathbf{C}_k^P$

The paired code means that the code indices $\{t_k\}$ and $\{f_k\}$ are one-to-one correspondence. In order words, the timing code index t_k is uniquely matched to the frequency code index f_k as follows:

$$\mathcal{I}_k^P = \left\{(t_k, f_k)\,|t_k \neq i_{T,l} \text{ and } f_k \neq i_{F,l}, \quad \forall k \neq l\right\}. \tag{2.30}$$

In this case, the maximum number of distinguishable code index pairs, $C_{\max}$, is $C_{\max} = \min\{C_T, C_F\}$, and the acquisition range for timing and frequency offset estimation is the same as the general acquisition case.

Table 2.1: Code Characteristics Comparison

Name	Codes $\mathbf{C}_k$	$\theta_{\max}$	$\lvert\epsilon_{\max}\rvert$	$C_{\max}$	$\mathcal{I}$ illustration
MUSIC	$e^{j2\pi k\frac{m}{M}}$	N	$\frac{N}{2N_T(M-1)}$	$M-1$	$\{(1,1),(2,2),(3,3)\}$
ESPRIT	$e^{j2\pi k\left(\frac{v}{V-1}+\frac{m}{M-1}\right)}$	$\frac{N}{V-1}$	$\frac{N}{2N_T(M-1)}$	$\min\{V,M\}-1$	$\{(1,1),(2,2),(3,3)\}$
$\mathrm{HR_P}$	$e^{j2\pi v\frac{t_k}{C_T}}e^{j2\pi m\frac{f_k}{C_F}}$	$\frac{N}{C_T}$	$\frac{N}{2N_TC_F}$	$\min\{C_T,C_F\}$	$\{(1,1),(2,3),(3,5),(4,4)\}$
$\mathrm{HR_T}$	$e^{j2\pi v\frac{t_k}{C_T}}$	$\frac{N}{C_T}$	$\frac{N}{2N_T}$	C_T	$\{(1,0),(2,0),(3,0),(4,0)\}$
$\mathrm{HR_F}$	$e^{j2\pi m\frac{f_k}{C_F}}$	N	$\frac{N}{2N_TC_F}$	C_F	$\{(0,1),(0,3),(0,5),(0,4)\}$
$\mathrm{HR_G}$	$e^{j2\pi v\frac{t_k}{C_T}}e^{j2\pi m\frac{f_k}{C_F}}$	$\frac{N}{C_T}$	$\frac{N}{2N_TC_F}$	C_TC_F	$\{(1,1),(1,3),(2,2),(2,3),(3,5),(4,4)\}$

Time code, $\mathbf{C}_k^T$

One of the big advantages of the proposed harmonic retrieval based code detection and offset estimation algorithm is that effective timing offset and frequency offset are automatically coupled based on Vandermonde product structure. Therefore, sometimes it is enough to use only one dimension to detect and estimate the code index. The code, $\mathbf{C}_k^T$, only uses the timing offset dimension to detect the codes. In this case, only t_k is used to determine all the codes, and the code index set is expressed as follows:

$$\mathcal{I}_k^T = \{(t_k,f_k)\,|f_k=0 \quad \forall k, \quad t_k \neq i_{T,l}, \quad \forall k\neq l\in[1,C_T]\}\,. \tag{2.31}$$

In this case, $C_{\max}$ is determined as C_T since the frequency dimension code index does not used for code detection. Since there is no code index in the frequency offset dimension, the acquisition range to estimate the frequency offsets can be extended to $|\epsilon_{\max}| < \frac{N}{2N_T}$.

Frequency code, $\mathbf{C}_k^F$

Similar to time code $\mathbf{C}_k^T$, the frequency code only utilize frequency offset dimension to estimate and detect the codes. For $\mathbf{C}_k^F$, the code index pair is given by

$$\mathcal{I}_k^F = \{(t_k,f_k)\,|t_k=0 \quad \forall k, \quad f_k \neq i_{F,l}, \quad \forall k\neq l\in[1,C_F]\}\,, \tag{2.32}$$

and the acquisition range of the timing offset can be extended up to $\theta_{\max} < N$

Grid code, $\mathbf{C}_k^G$

In this case, it is allowed to use the same code index t_k or f_k to increase the number of distinguishable code index pairs $C_{\max}$. The general code index pair can be described as

$$\mathcal{I}_k^G = \{(t_k,f_k)\,|(t_k,f_k)\neq(i_{T,l},i_{F,l}), \quad \forall k\neq l\in[1,C_{\max}]\}\,. \tag{2.33}$$

The overall parameters are summarized in Table 2.1

2.5 Computational Complexity

In terms of computational complexity, basically multi-dimensional HR algorithm has the disadvantages compared to 1-D MUSIC and 1-D ESPRIT algorithms. However, since multi-dimensional resource is simultaneously utilized to detect the code and to estimate the offsets, more nodes can be supported with limited resource and more accurate estimation and detection can be performed. The overall computational complexity of each algorithm is summarized in the following:

Table 2.2: Computational Complexity of the HR Algorithm

Operation	Dimension Size	Required Flops
SVD of $\tilde{\mathbf{X}}$ to compute $\mathbf{U}_s$	$V_1M_1 \times 2QV_2M_2$	$4(2QV_2M_2)^2V_1M_1 + 13(V_1M_1)^3$
SVD to compute $\mathbf{U}_1^\dagger$	$(V_1-1)M_1 \times K$	$4((V_1-1)M_1)^2K + 8(V_1-1)M_1K^2 + 9K^3$
Multiplication of $\mathbf{U}_1^\dagger\mathbf{U}_2$	$K \times K$	$(V_1-1)M_1K^2$
QR EVD of $\mathbf{U}_1^\dagger\mathbf{U}_2$	$K \times K$	$25K^3$
Multiplication of $\mathbf{U}_s\mathbf{L}$	$V_1M_1 \times K$	$V_1M_1K^2$
Estimate parameters of $e^{j2\pi\eta_k}$	$V_1M_1 \times K$	$(V_1-2)M_1F$
Estimate parameters of $e^{j2\pi\xi_k}$	$V_1M_1 \times K$	$(V_1-1)(M_1-1)F$

Table 2.3: Computational Complexity of the ESPRIT Algorithm

Operation	Dimension Size	Required Flops
Compution of sample covariance matrix, $\mathbf{R}_V$	$V \times V$	MV^2Q
QR EVD of R_V	$V \times V$	$25V^3$
SVD to compute $\mathbf{U}_1^\dagger$	$(V-1) \times K$	$4(V-1)^2K + 8(V-1)K^2 + 9K^3$
Multiplication $\mathbf{U}_1^\dagger\mathbf{U}_2$	$K \times K$	$(V-1)K^2$
QR EVD of $\mathbf{U}_1^\dagger\mathbf{U}_2$	$K \times K$	$25K^3$
Compution of sample covariance matrix, R_M	$M \times M$	M^2VQ
QR EVD of R_M	$M \times M$	$25M^3$
SVD to compute $\mathbf{U}_1^\dagger$	$(M-1) \times K$	$4(M-1)^2K + 8(M-1)K^2 + 9K^3$
Multiplication $\mathbf{U}_1^\dagger\mathbf{U}_2$	$K \times K$	$(M-1)K^2$
QR EVD of $\mathbf{U}_1^\dagger\mathbf{U}_2$	$K \times K$	$25K^3$

Table 2.4: Computational Complexity of the MUSIC Algorithm

Operation	Dimension Size	Required Flops
Compution of sample covariance matrix, R_M	$M \times M$	M^2VQ
QR EVD of R_M	$M \times M$	$25M^3$
Line search for cost function $\Psi_k(\epsilon)$	$E \times 1$	$EM^3(M-K)^2$
Matrix calculation $\mathbf{C}^H(\epsilon)\mathbf{C}(\epsilon)$	$K \times K$	MK^2
Inverse of $\mathbf{C}^H(\epsilon)\mathbf{C}(\epsilon)$	$K \times K$	$25K^3$
Multiplication $[\mathbf{C}^H(\epsilon)\mathbf{C}(\epsilon)]^-1\mathbf{C}^H(\epsilon)$	$K \times K$	K^3
Timing offset estimation		VQK

2.6 Simulation Results

The overall system model and simulation parameters are based on the environment in [7]. The total bandwidth is 10MHz, and $N = 1024$. Let us assume that each ranging subchannel is composed of $Q = 3$ tiles with $V = 6$ consecutive subcarriers, while $M = 5$ OFDMA blocks are presented in ranging subchannels. The ITU vehicular A channel model is applied to evaluate the performance [12].

The timing and frequency offset ranges are designed for $\theta_k \in [0, 114)$ and $\epsilon_k \in [-0.02, 0.02]$ which correspond to the system with 1.5 km cell radius and 100 km/h Doppler frequency at $f_c = 2.4$ GHz. The timing and frequency ranges of the proposed method are $\theta_{\max} < N/D_T$ and $|\epsilon_{\max}| < N/(2N_T D_F)$, and it can be adaptively adjusted by designing the parameter value D_T and D_F. In the simulation setup, we choose $D_T = D_F = 6$.

To verify the superiority of the proposed algorithm in terms of the code detection and offset estimation performance, the ranging algorithms using MUSIC [5] and ESPRIT [6] are also compared under a common simulation setup. In addition, the very recent ranging algorithm exploiting Zadoff-Chu sequence structure [13] [14] is also compared as a reference under almost same environment. The overall simulation parameters and frame structure are shown in Table 2.5 and Fig. 2.3

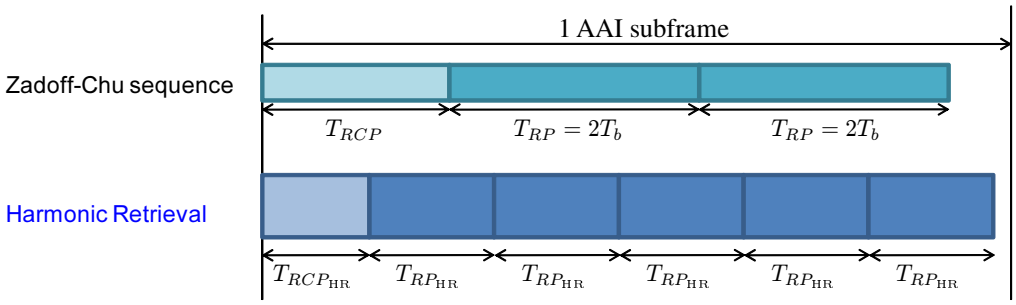
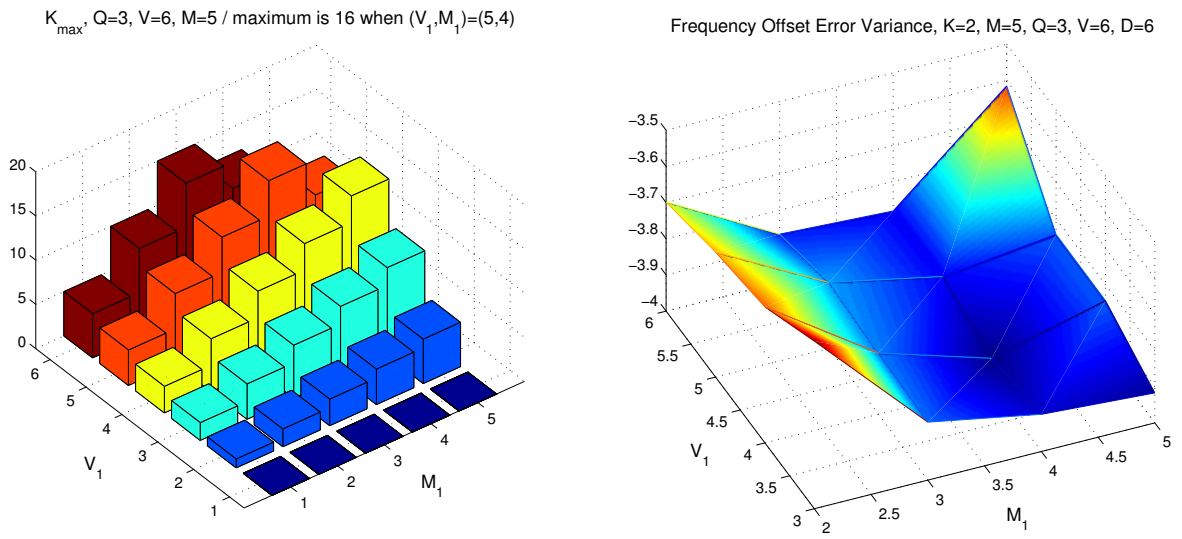


Figure 2.3: Transmitted frame structure for comparison between harmonic retrieval (HR) and Zadoff-Chu (ZC) sequence algorithms

Table 2.5: Simulation Setup

Parameters	HR, MUSIC, ESPRIT	Zadoff-Chu Sequence
Total # of subchannels	$R = 4$	1
# of tiles / subchannel	$Q = 3$	1
# of subcarriers / tile	$V = 6$	72 (oversampled as 144)
Tot. # of subcarriers	$RQV = 72$	72 (oversampled as 144)
# of active users	8, 12 ($K = 2, 3$)	8, 12
# of maximum users	16 (MUSIC, ESPRIT), 24 (HR)	24
# of symbols	$M = 5$	2 (by oversampling)
Frame length	$5.875T_b$	$\approx 5.5T_b$
Max. timing offset	114	114
Max. freq. offset	[-0.02,0.02]	[-0.04,0.04]



(a) $K_{\max}$ according to the smoothing factor (V_1, V_2, M_1, M_2)

(b) Frequency estimator error variance according to the smoothing factor

Figure 2.4: Smoothing factor decision. By adjusting (V_1, V_2), and (M_1, M_2), the maximum number of distinguishable RSSs $K_{\max}$ (Fig. 2.4a) and the performance of timing and frequency offset estimation (Fig. 2.4b) is changed, so it is required to find the best (V_1, V_2), and (M_1, M_2) combination matched to the purpose.

2.6.1 Smoothing factor decision

Let us recall that the smoothing operator $\mathcal{S}(\mathbf{x})$ enlarges the observed data matrix $\mathbf{X}_q \in \mathbb{C}^{V \times M}$ to the smoothed data matrix $\mathbf{X}_{q,\mathcal{S}} \in \mathbb{C}^{V_1 M_1 \times V_2 M_2}$ by choosing appropriate smoothing factors V_1, V_2 and M_1, M_2 with the condition $V_1 + V_2 = V + 1$ and $M_1 + M_2 = M + 1$. Therefore, the size of smoothed data matrix $\mathbf{X}_{q,\mathcal{S}}$ depends on the smoothing factors. In fact, this smoothing factor affects the code detection and offset estimation performance as well as the maximum number of distinguishable RSSs.

As shown in (2.27), $K_{\max}$ is determined by (V_1, V_2) and (M_1, M_2) pair values, and Fig. 2.4a shows the variation of this $K_{\max}$ according to V_1 and M_1 for given $Q = 3$, $V = 6$ and $M = 5$. From (2.27) and Fig. 2.4a, we can choose the best smoothing factor combination to support the maximum number of RSSs for given sample matrix, e.g., $K_{\max} = 16$ with (V_1, M_1)=(5,4).

Fig. 2.4b shows the error variance performance of frequency offset estimation according to the smoothing factor. In the similar way, the performance of code detection and offset estimation is varied with respect to the smoothing factor. It is because the size of the smoothed matrix, which determine the code detection and offset estimation, depends on the smoothing factor. Consequently, the smoothing factor can be also chosen to show the better performance. However, the best smoothing factor combination is different from each other depending on what the performance metric is, such as K_max or error variance minimization. In our simulation setup, we picked up the smoothing factor as $(V_1, M_1) = (4, 4)$ which shows good performance in terms of frequency offset estimation as well as keeps the comparable performance in terms of $K_{\max}$ and the other performance metric.

2.6.2 Miss-detection probability

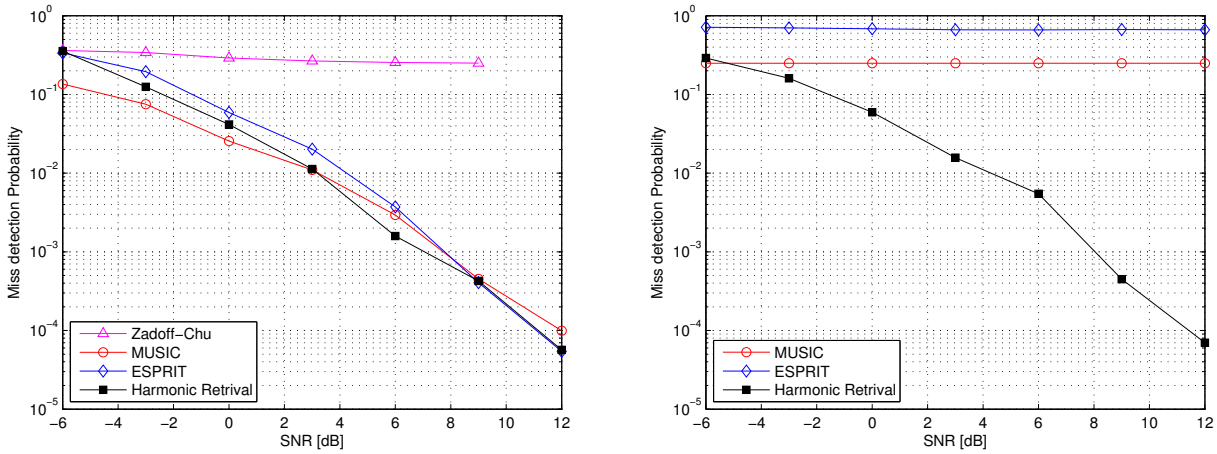
Table 2.6: Code detection and offset estimation notation

Main algorithm	Code detector	Timing offset estimator	Frequency offset estimator
Zadoff-Chu (ZC)	**ZCCD** (ZC Code Detector)	**ZCTE** (ZC Timing Estimator)	**ZCFE** (ZC Frequency Estimator)
MUSIC	**MCD** (MUSIC Code Detector)	**AHTE** (Adhoc Timing Estimator)	**MFE** (MUSIC Frequency Estimator)
ESPRIT	**ECD** (ESPRIT Code Detector)	**ETE** (ESPRIT Timing Estimator)	**EFE** (ESPRIT Frequency Estimator)
Harmonic Retrieval (HR)	**HRCD** (HR Code Detector)	**HRTE** (HR Timing Estimator)	**HRFE** (HR Frequency Estimator)

Fig. 2.5 describes the missing probability for four code detection algorithms with respect to signal-to-noise ratio (SNR) for $K = 2$ and $K = 5$. The missing probability is defined as the event which the ranging codes of RSSs can not be distinguished at the BS including *code collision* events. The notation for each code detector and offset estimator is expressed in the Table 2.6.

As seen in the figures, the proposed HRCD shows good P_{miss} performance. It is because auto pairing property in HR improves the performance of code indices detection. Even though ECD also exploits two independent estimates of the code indices to make a decision, the mis-detection probability because it is of frequent occurrence for ECD not to coincide with the same code indices for two independent estimations.

In addition, as the number of ranging users K increases (as shown in Fig. 2.5b), the performances of ECD and MCD rapidly worsen even though error event probabilities for all methods are similar. It is because ECD and MCD only support $\min(K, M) - 1$ ranging users, respectively whereas HRCD can support up to $K_{\max} \leq \min((V_1 - 1)M_1, 2QV_2M_2)$ almost surely as mentioned before in (2.27). Since the time-frequency offset information is jointly exploited and harmonized to estimate the code indices, the resolvable number of users increases in HRCD. Table 2.7 shows the maximum number of resolvable users for specific M and V. As seen in Table 2.7, the RSS capability of HRCD continuously increases as either the number of subcarriers or symbols increases.



(a) P_{miss} when $K = 2$

(b) P_{miss} when $K = 5$

Figure 2.5: P_{miss} miss probability comparison, $(Q, V, M, D) = (3, 6, 5, 6)$, For Harmonic Retrieval method, $(V_1, V_2) = (4, 3)$, $(M_1, M_2) = (4, 2)$ is applied as a smoothing factor.

Table 2.7: $K_{\max}$ variation

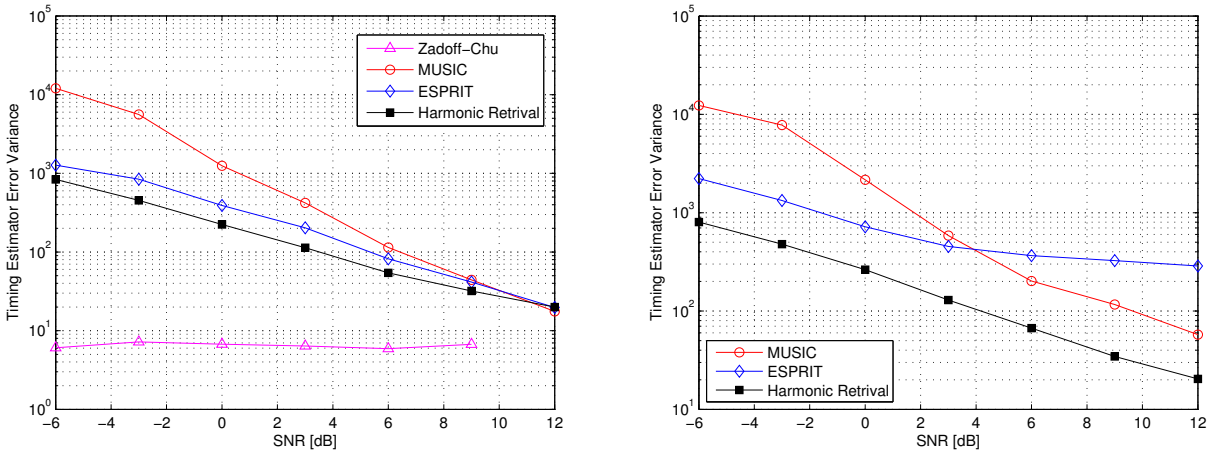
$K_{\max}$	Values (M, the number of OFDMA symbols)							
	2	3	4	5	6	7	8	9
MUSIC,ESPRIT	1	2	3	4	5	6	7	8
Harmonic Retrieval	8	12	12	16	20	24	24	28
$K_{\max}$	Values (V, the number of subcarriers)							
	3	4	5	6	7	8	9	10
MUSIC,ESPRIT	2	3	4	4	4	4	4	4
Harmonic Retrieval	8	12	12	16	20	24	24	28

2.6.3 Timing and frequency offset estimation accuracy

Figs. 2.6 and 2.7 show the error variance performances of the timing and frequency offset estimation with respect to different number of active RSS users in the system. As seen in the figures, both HRTE and HRFE show good performance regardless of the number of active users, K. But, in Fig. 2.6a, the error variance of ZCTE is much smaller than HRTE. Since the code detection and parameter estimation is tightly coupled in the ranging scenario, we have to think about the code detection as well as the estimation performance. The parameter estimation performance in Figs . 2.6 and 2.7 only cover the RSSs which are successfully detected at the BS. Since the code detection performance of ZCCD is very poor, only few codes having good characteristics can be detected at the BS. Since the timing offset estimation is inherent in the code detection by code correlation sliding in the time domain, it is obvious to have small error variance due to short range of sliding window.

On the other hand, the other algorithms including HRTE have both timing offset and code index in the phase rotation, therefore, wrong code detection causes a big timing offset estimation error.

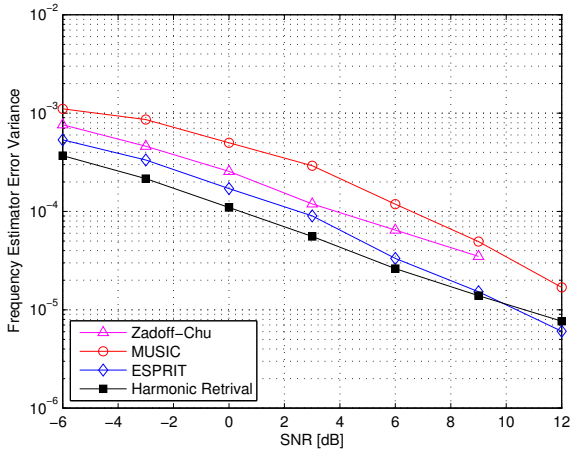
However, almost RSSs can not be detected at the BS for ZCCD, small error variance with respect



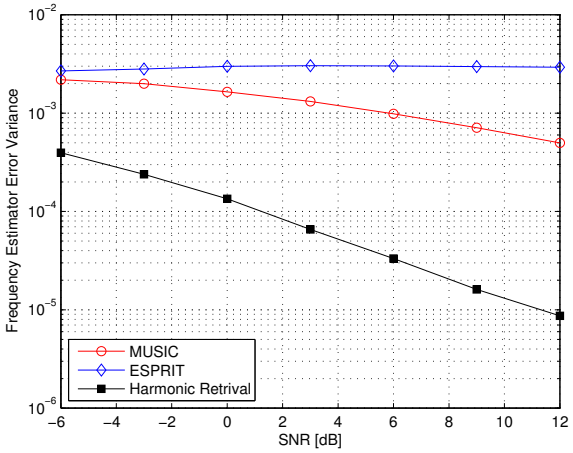
(a) Timing Estimator Error Variance, $K = 2$

(b) Frequency Estimator Error Variance, $K = 5$

Figure 2.6: Error variance of timing offset estimation comparison for different number of users, $K = 2$ or 5.



(a) Frequency Estimator Error Variance, $K = 2$



(b) Frequency Estimator Error Variance, $K = 5$

Figure 2.7: Error variance of frequency offset estimation comparison for different number of users, $K = 2$ or 5.

to timing offset does not have big advantage compared to other algorithms.

As the number of users increases, the performance of other algorithms undergo severe performance degradation because wrongly detected code index disturbs timing and frequency offset estimation. However, the proposed HRTE and HRFE algorithm is more robust to the wrongly detected code effect.

It is because both the timing and frequency code indices are automatically paired together, so more strict "detected code verification" process can be applied to detect the code as well as estimate the offsets.

2.7 Summary and Key Advantages of Harmonic Retrieval based Ranging Process

- Flexible subcarrier allocation usage
  - The proposed algorithm can be applied for any type of subcarrier allocation based on small *tile* structure.
  - Zadoff-Chu sequence algorithm should use big sub-band allocation (at least 72 consecutive subcarriers in our simulation setup) whereas the other algorithms only require 6 consecutive subcarriers to operate the ranging process.

- Efficient data usage to distinguish multiple RSSs.
  - The proposed algorithm fully utilize the observed data matrix to distinguish multiple RSS users as many as possible.
  - In order to increase the maximum number of RSS users, $K_{\max}$, MUSIC and ESPRIT should increase the number of OFDMA symbols as well as the number of subcarriers because $K_{\max} =$

$\min(M-1.V-1)$. It means that many OFDMA symbols should be repeated and many consecutive subcarriers should be allocated simultaneously. If one of the parameters is fixed, there is no way to increase $K_{\max}$ even though the increase of the other parameter.

  - Even in the practical scenario which only a few OFDMA symbols are utilized in the ranging process, the proposed HR algorithm can increase $K_{\max}$ by adjusting the number of tiles Q and the number of subcarriers V. Therefore, the HR algorithm smartly utilizes the given data matrix to distinguish many RSS users simultaneously.

- Good code detection and offset estimation performance
  - The proposed algorithm outperforms the other algorithms in terms of the code detection and offset estimation as well as supports more number of RSS users.

2.8 Conclusion

The improved parameter estimation and pairing algorithm is proposed for initial ranging process in OFDMA uplink systems. The proposed method using HR technique enables the BS to pair up ranging codes, timing and frequency offsets simultaneously, and it eliminates the multiple parameter coordination problems in initial ranging process. These simulation results verify the significantly increased users identification capability without sacrificing the estimation performance.

Chapter 3

Timing Adjustment Techniques to Mitigate Interference between Multiple Nodes in OFDMA Mesh Networks

We configure the multiple node interference (MNI) on OFDMA mesh networks and analyze the feature of this MNI as a closed form in terms of timing misalignment between multiple nodes. Based on our analysis, we propose the new timing adjustment techniques to mitigate MNI and verify the difference and superiority of the proposed techniques compared to the ones typically used in cellular OFDMA systems.

3.1 Introduction

Recently, wireless mesh networks have emerged as a promising technology for next-generation wireless networks [15] [16] with rapid increase on the demand for higher-speed Internet access from anywhere, which requires high data rate, quality-of-services(QoS) support and enlarged coverage with reliable connectivity. Especially, taking advantage of OFDMA systems which granularly utilize the resources, extensive studies are performed over OFDMA mesh networks to realize the potential of this new technology [15]. However, because of the challenges and difficulties presented by the multihop communications over OFDMA mesh networks, OFDMA air interface is not configured as a practical network model [17].

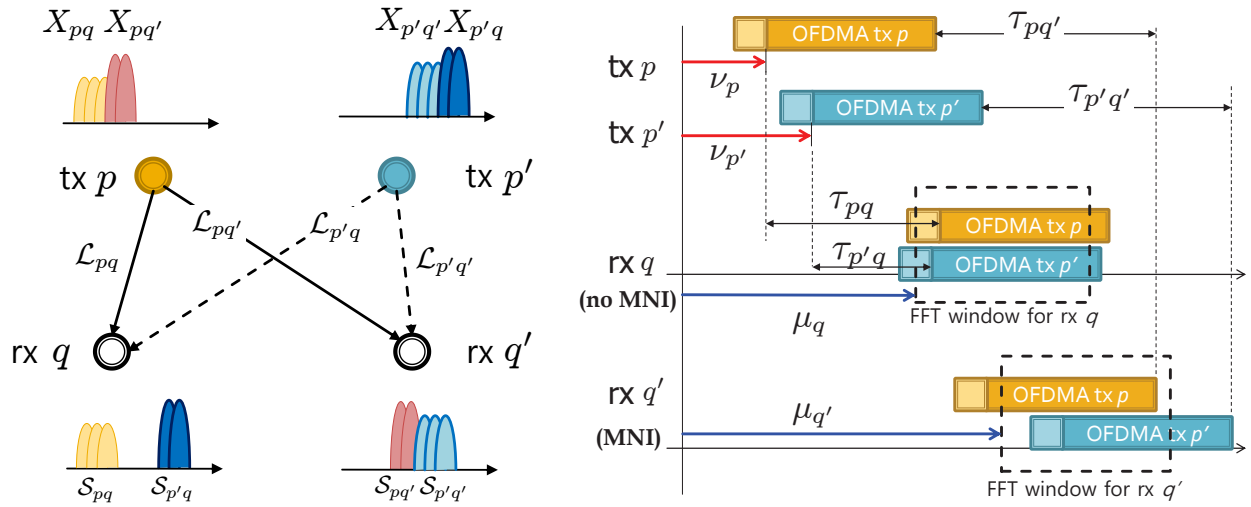
Although lots of researches are performed on OFDMA mesh network to resolve these problems, those researches are mainly focused on MAC layer such as resource allocation, scheduling, while excluding PHY layer issues such as node synchronization, interference management, and so on. However, the researches on PHY layer issues are also essential to configure OFDMA mesh network in practice.

Consequently, we undertake PHY layer issues on OFDMA mesh network, especially the synchro-

nization challenges and interference features in this chapter. The main goal of this chapter is twofold: First, we define multiple node interference (MNI) on OFDMA mesh network, which is caused by timing and frequency misalignments between multiple nodes. Then, we quantify the effects of this misalignment on MNI under OFDMA mesh networks with an analytic signal-to-interference -noise ratio (SINR) expression. Next, we clarify the main difference on timing adjustment strategy between cellular based OFDMA systems (e.g., downlink/uplink) and OFDMA mesh network, and propose the timing adjustment criterion to alleviate the performance degradation on mesh networks.

The remainder of this chapter is organized as follows: Section 3.2 describes OFDMA mesh network model considered in this chapter and presents the analytic MNI features with respect to timing difference between multiple nodes. We explain the limit of existing adjustment methods and propose new timing adjustment techniques to mitigate SINR degradation due to MNI in Section 3.3, then observe the performance of OFDMA mesh networks in Section 3.4. Section 3.5 concludes this chapter.

3.2 System model



(a) Frequency domain, The symbol X_{pq} is delivered via the link $\mathcal{L}_{pq}$ using subcarriers in $\mathcal{S}_{pq}$

(b) Time domain, There is propagation delay τ_{pq} on the link $\mathcal{L}_{pq}$. ν_p and μ_q denote tx and rx *timing adjustments*

Figure 3.1: OFDMA mesh network example, two tx nodes and two rx nodes communication scenario

Let us consider a generalized OFDMA wireless mesh network employing a total of N subcarriers, where P transmitting nodes communicate with Q receiving nodes simultaneously. Unlike cellular-type of point-to-multipoint (PMP) topology, mesh mode makes each node not only a host but also a router that forwards packets on behalf of others. Consequently, each node can transmit data to multiple nodes as well as receive data from multiple nodes.

Figure 3.1 illustrates the OFDMA mesh network scenario considered here. As already declared, the entire network is grouped into transmitting node set $\mathcal{N}_p = \{1, 2, \ldots, P\}$ and receiving node set $\mathcal{N}_q = \{1, 2, \ldots, Q\}$. Since each node can be a host as well as a router in mesh network, transmitter(tx) p contains different data delivering to different destinations, denoted as $X_{pq}, \forall q \in \mathcal{N}_q$ which is the data

delivered via the link $\mathcal{L}_{pq}$ from tx p to receiver(rx) q.

In order to avoid the subcarrier collision, it is suggested to allocate the subcarriers exclusively. Let us define $\mathcal{S}_p$ as the subcarrier index group assigned to the node tx p, and this group is divided into Q subgroups, named $\mathcal{S}_{pq}$, which includes subcarrier indices used at the link $\mathcal{L}_{pq}$: $\mathcal{S}_p = \cup_{q=1}^{Q} \mathcal{S}_{pq}$ where $\mathcal{S}_{pq} \cap \mathcal{S}_{p'q'} = \emptyset, \forall p \neq p' \in \mathcal{N}_p, q \neq q' \in \mathcal{N}_q$.

As shown in Figure 3.1b, depending on the propagation delay and timing adjustment parameters, MNI may be reduced at the receiving node (e.g., rx q) or may become serious (e.g., rx q'). Consequently, in order to prevent or mitigate MNI at the mesh network, at first we should estimate the timing and frequency offsets for each link, and then apply the advanced adjustment strategy to alleviate the performance.

The following equation describes the received symbol at rx q with timing offset over an OFDMA mesh network.

$$y_q[n] = \sum_{p=1}^{P} h_{pq}[n] * x_p[n - \nu_p - \tau_{pq}], -N_q \leq n \leq N - 1$$

where $*$ denotes the convolution and h_{pq} means the channel impulse response on link $\mathcal{L}_{pq}$, which connects tx p to rx q. N_g is the length of cyclic prefix, which is assumed to be longer than the maximum channel delay spread $L - 1$, $x_p[n]$ is the transmitted symbol from tx p and τ_{pq} means the propagation delay on link $\mathcal{L}_{pq}$. In addition, ν_p means *tx timing adjustment* parameter at the tx p for synchronization (see Figure 3.1b).

The received symbol at the frequency domain is given by

$$Y_q[k] = \sum_{n=0}^{N-1} y_q[n + \mu_q] e^{j2\pi nk/N},$$

where μ_q denotes *rx timing adjustment* parameter at the rx q for synchronization. Therefore, the *effective timing delay* at node rx q on link $\mathcal{L}_{pq}$ is denoted as

$$\theta_{pq} = \nu_p + \tau_{pq} - \mu_q. \tag{3.1}$$

3.2.1 Interference Features

On OFDMA uplink systems, the multiple access interference (MAI) is only expressed by the timing and frequency offset mismatch between multiple subscriber stations (SSs) since all the SSs share one common base station (BS) to communicate [18] [19].However, all tx nodes communicate with multiple rx nodes simultaneously in the OFDMA mesh network. Thus timing and frequency offset adjustment for one specific link can affect all the other connected links in the network. Therefore, the MNI should be expressed based on the link parameters between tx and rx nodes in the OFDMA mesh network.

Let us define the communication link between tx p and rx q, i.e., link $\mathcal{L}_{pq}$, as the desired link at the receiving node rx q. Then, the interference observed at the node rx q can be classified into four classes as follows:

- ***Self-transmitting Self-receiving Node Interference* (SSNI):** the interference caused by timing misalignment on the desired link $\mathcal{L}_{pq}$. SSNI may or may not occur. Precisely, there is no SSNI

Table 3.1: OFDMA mesh network interference classification

Variables	SSNI	SONI	OSNI	OONI
Link	$\mathcal{L}_{pq}$	$\mathcal{L}_{pq'}$	$\mathcal{L}_{p'q}$	$\mathcal{L}_{p'q'}$
Concerned offset parameters	$v_p + \tau_{pq}$ μ_q	$v_p + \tau_{pq}$ μ_q	$v_p + \tau_{pq}$ $v_{p'} + \tau_{p'q}$ μ_q	$v_p + \tau_{pq}$ $v_{p'} + \tau_{p'q}$ μ_q
Subcarriers	$\mathcal{S}_{pq}$	$\mathcal{S}_{pq}, \mathcal{S}_{pq'}$	$\mathcal{S}_{pq}, \mathcal{S}_{p'q}$	$\mathcal{S}_{pq}, \mathcal{S}_{p'q'}$
Example	X_{pq}	$X_{pq'}$	$X_{p'q}$	$X_{p'q'}$

for $0 \leq \theta_{pq} \leq N_g - L + 1$.

- ***Self-transmitting Other-receiving Node Interference*** **(SONI):** the interference caused by timing misalignment on the link $\mathcal{L}_{pq'}$. SONI may occur when the node tx p also delivers the data to the other node rx q'.
- ***Other-transmitting Self-receiving Node Interference*** **(OSNI):** the interference caused by timing misalignment on the link $\mathcal{L}_{pq'}$.
- ***Other-transmitting Other-receiving Node Interference*** **(OONI):** the interference caused by timing misalignment on the link $\mathcal{L}_{pq'}$.

This is generalized MNI classification for OFDMA mesh network. Therefore, depending on subcarrier allocation strategy and link connectivity [16], the MNI term can be described for various kinds of OFDMA mesh network. The detail parameters to dominate the interference is classified in Table 3.1.

3.2.2 Received Signal to Interference Noise Ratio (SINR)

Based on the interference classification, the received SINR at rx q on the k-th subcarrier ($k \in \mathcal{S}_{pq}$) is expressed as

$$\mathrm{SINR}_q[k] = \frac{E_q^2[k]}{\sigma_{q,\mathrm{SSNI}}^2[k] + \sigma_{q,\mathrm{SONI}}^2[k] + \sigma_{q,\mathrm{OSNI}}^2[k] + \sigma_{q,\mathrm{OONI}}^2[k] + \sigma_n^2}. \tag{3.2}$$

where $E_q^2[k]$ is the desired signal power at the k-th subcarrier, and $\sigma_{q,\mathrm{SSNI}}^2[k]$, $\sigma_{q,\mathrm{SONI}}^2[k]$, $\sigma_{q,\mathrm{OSNI}}^2[k]$, and $\sigma_{q,\mathrm{OONI}}^2[k]$ denote the variances of SSNI, SONI, OSNI, and OONI observed at the k-th subcarrier, respectively. σ_n^2 is the noise variance. The detail derivations are shown in the Appendix.

Let us define $\gamma_{mk}(n_L, n_U)$ as the *normalized orthogonal sufficiency* to evaluate the orthogonality degradation caused by timing and frequency offset misalignment as follows:

$$\gamma_{mk}(n_L, n_U) = \frac{1}{N} \sum_{n=n_L}^{n_U} e^{\frac{j2\pi n(m-k+\epsilon_a-\epsilon_b)}{N}} u(n_U - n_L) \tag{3.3}$$

where $u(n)$ is unit step function, $k \in \mathcal{S}_{aq}$ and $m \in \mathcal{S}_{pb}$.

3.3 Timing adjustment for MNI mitigation

3.3.1 Tx Node Timing Adjustment Strategy

Typically in OFDMA uplink systems, the base station (BS) (rx) forwards timing adjustment feedback to each subscriber stations (txs) in order to adjust timing offset and align at the BS [10]. However, it is very difficult to apply this timing adjustment strategy for OFDMA mesh network since the timing adjustment for specific rx node can cause the timing misalignment for all the other rx nodes.

If it is highly required to use *tx timing adjustment* in the system, then the artificial timing adjustment at tx p, ν_p, should satisfy the following condition in order to eliminate all the MNI for given propagation delays, τ_{pq}'s:

$$\max_p(\theta_{pq}) - \min_p(\theta_{pq}) \leq N_g - L + 1, \forall q \in \mathcal{N}_q \tag{3.4}$$

In order to obtain ν_p's to satisfy the condition (3.4), each tx node should know all τ_{pq}'s as well as all the other tx nodes timing adjustment ν_p's. Therefore, the central center should be required to collect all the information, calculate the proper ν_p's and distributed the achieved parameters, but it requires too much complexity and feedback burden. Moreover, there are some unaffordable cases not to obtain the proper ν_p's depending on the given τ_{pq}'s. Consequently, the timing adjustment method used in OFDMA uplink system is no longer attractive any more in order to mitigate the MNI over OFDMA mesh network.

3.3.2 Rx Node Timing Adjustment Strategy

Unlink OFDMA downlink systems, it is addressed in OFDMA mesh network that the perfect synchronization for rx q, i.e., $\mu_q = \tau_{pq}$, does not guarantee the best performance any more since the desired link $\mathcal{L}_{pq}$ still can be disturbed by other links $\mathcal{L}_{p'q}, \forall p' \in \mathcal{N}_p$ because of the timing misalignment between links. There are other synchronization points to enable better performance according to the distribution of timing offset. Therefore, even though we estimate very accurate timing offsets for all links, it should be performed finding the best timing adjustments μ_q, which is not the exact timing offset of the desired link but the compromised offset from all the links, to minimize MNI. Using the derived analytic SINR expression in (3.2), we propose two *rx timing adjustment* strategies to improve the performance in the following.

Maximize Average SINR

$$\mu_{q,\text{avg}} = \arg\max_{\mu_q} \frac{1}{|\mathcal{S}_{pq}|} \sum_{k \in \mathcal{S}_{pq}} \text{SINR}_q[k] \tag{3.5}$$

Since $\text{SINR}_q[k]$ is the function of μ_q, the best timing adjustment parameter can be obtained by using the analyzed SINR expression. Intuitively, this method is applied to maximize the average capacity on the link $\mathcal{L}_{pq}$.

Maximize Minimum SINR

$$\mu_{q,\text{maxmin}} = \arg\max_{\mu_q} \min_{k \in \mathcal{S}_{pq}} \text{SINR}_q[k] \tag{3.6}$$

This adjustment technique picks up μ_q which maximize the minimum SINR. This strategy is appropriate to minimize the average bit error rate (BER) on the link $\mathcal{L}_{pq}$ since BER is dominantly determined by the weakest SINR.

3.4 Simulation results

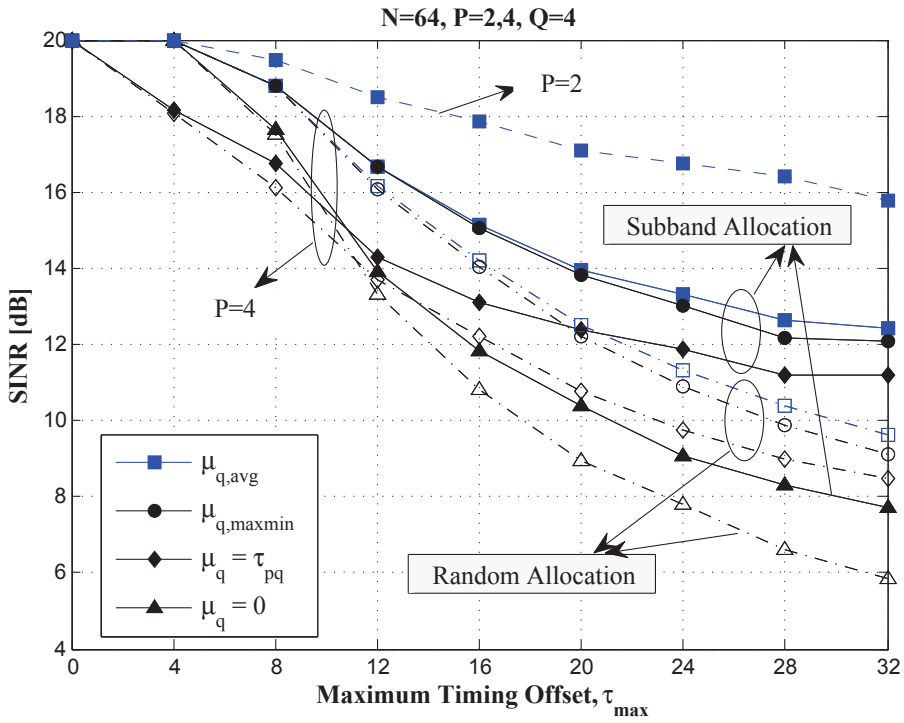
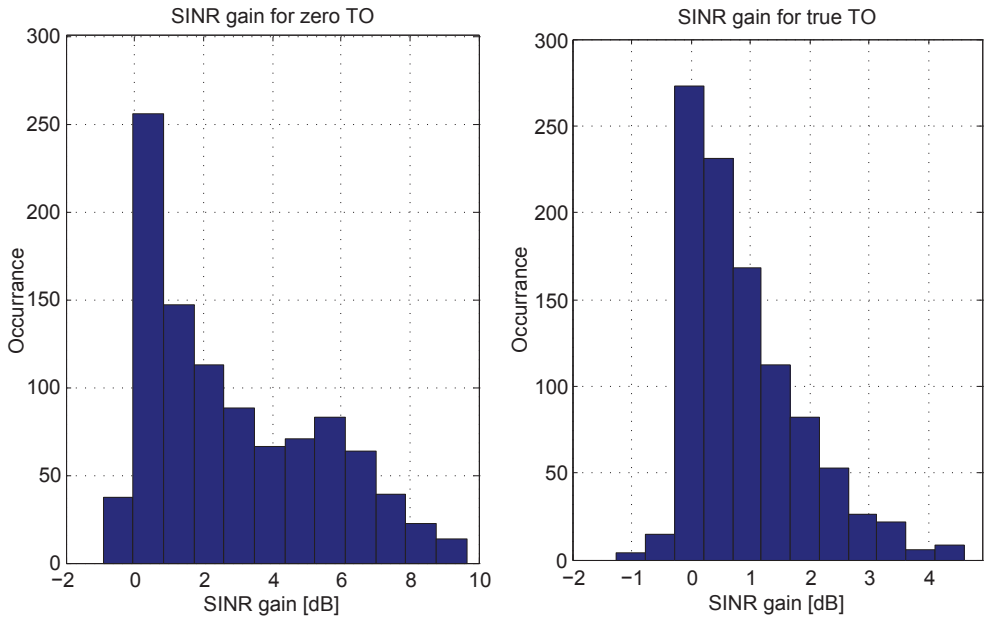


Figure 3.2: SINR comparison with respect to maximum propagation delay, τ_{pq}

We evaluate the performance of the proposed rx timing adjustment techniques in terms of SINR. Let us assume the OFDMA system parameters as $N = 64, N_g = 8, L = 4$ and mesh network with $P = 2, 4$ and $Q = 4$ nodes.

3.4.1 SINR performance

Figure 3.2 shows the SINR performance of *rx timing adjustment* methods with respect to the maximum timing offsets where $\tau_{pq} \in [0, \tau_{\max}]$. As shown in the figure, the proposed timing adjustment techniques $\mu_{q,\text{avg}}$ and $\mu_{q,\text{maxmin}}$ outperform both timing adjustment technique using true timing offset $\mu_q = \tau_{pq}$ and no timing adjustment $\mu_q = 0$ cases. In addition, it is observed that the SINR improvement using the proposed adjustment techniques is more prominent as $\tau_{\max}$ increases. It means the usage of the proposed adjustment are more important for the system with large network size.



(a) SINR gain for Zero TO (b) SINR gain for True TO

Figure 3.3: relative gain of $\text{SINR}_q[k]|_{\mu_{q,avg}}$, $\tau_{\max} = 16$.

In addition, we compare the SINR performance for different subcarrier allocation schemes. Since the MNI feature is quite different depending on what the allocation scheme is used in the mesh network, it is also important to allocate subcarriers smartly in order not to cause too much MNI. As shown in the figure, basically exclusive subband allocation scheme shows better SINR performance compared to random allocation since only edge subcarriers are dominantly sacrificed by interference invasion on subband allocation [19]. Moreover, even though it applies the complex subcarrier and power allocation scheme appropriate to OFDMA mesh network [16], the proper timing adjustment parameters can be obtained by exploiting analyzed SINR expression ((3.2), (3.11)-(3.15)) on mesh topology, which is differentiated from cellular OFDMA uplink MAI feature. In addition, as the number of active nodes is increased from $P = 2$ to $P = 4$, it is shown that the SINR performance is severely decreased. It means that, in the heavy network, it is more crucial to find the best $\mu_{q,avg}$ parameters to mitigate MNI.

Figure 3.3 shows the relative SINR gain of the proposed technique compared to zero ($\mu_q = 0$) and true ($\mu_q = \tau_{pq}$) timing compensation methods. As shown in the figure, by adopting $\mu_{q,avg}$, we can achieve significant gain in terms of SINR for almost cases.

3.4.2 Error Probability Performance

Figure 3.4 describes the BER performance of the desired receiver according to the maximum timing offset ($\tau_{\max}$). As depicted in the figure, the BER performance of the proposed timing adjustment technique outperforms the other methods since the appropriate timing adjustment parameter μ_q is chosen to minimize MNI, and the performance difference becomes bigger as the maximum timing offset increases, which means that the proposed timing adjustment $\mu_{q,\text{avg}}$ can maintain the performance by

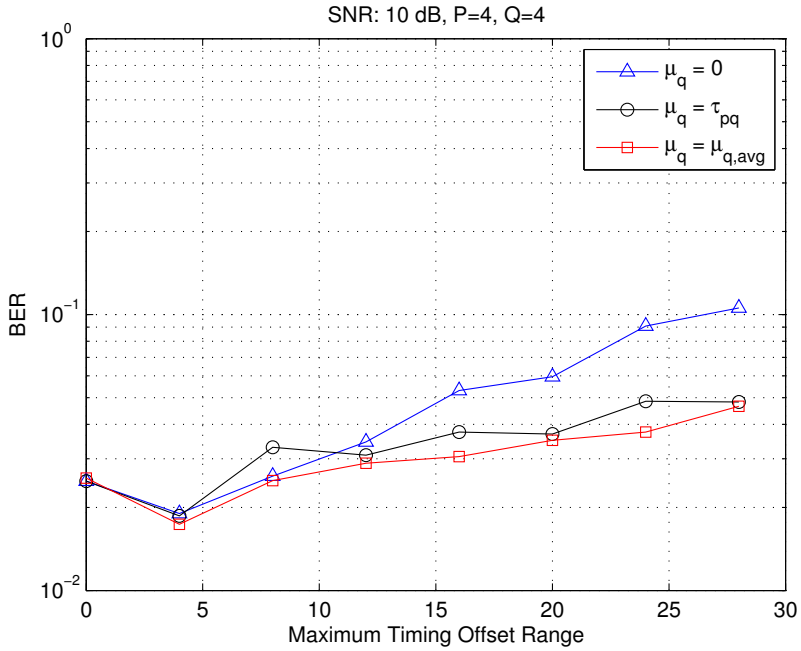


Figure 3.4: BER performance according to the maximum timing offset τ_{pq}, SNR = 10dB

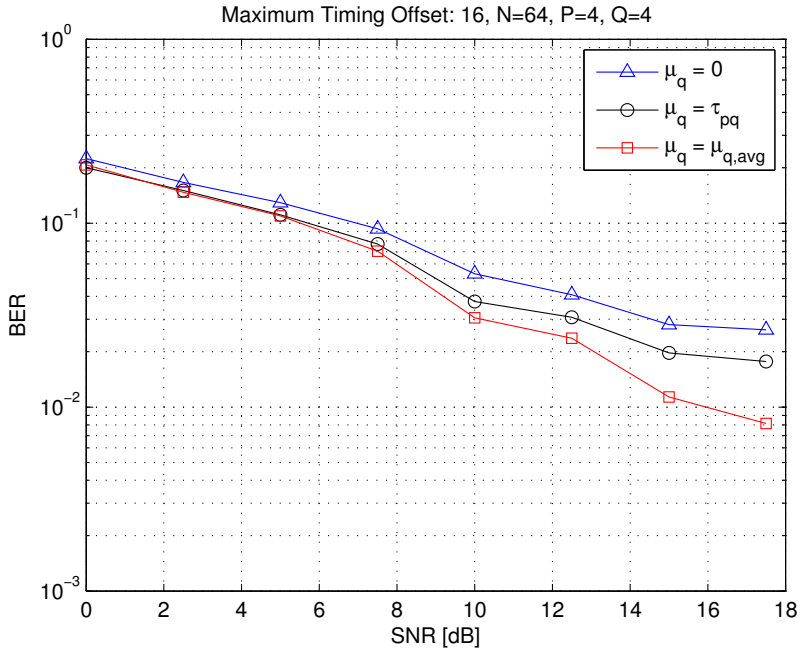


Figure 3.5: BER performance according to SNR of the desired pair, $\tau_{\max} = 16$

alleviating the degradation caused by MNI. In addition, the proposed adjustment technique can be usefully utilized for the large timing offset circumstance such as the system with large cell size or big propagation delays.

Figure 3.5 shows the BER performance according to the SNR of the desired pair when maximum timing offset is fixed as $\tau_{\max} = 16$. As shown in the figure, the proposed adjustment method becomes

much better for interference-limited environment (high SNR circumstance) since appropriate timing adjustment μ_q is suggested at the receiver.

3.5 Conclusion

We declare four different kinds of MNI in OFDMA mesh network and analyze the feature of MNI in terms of timing and frequency misalignment. Unlike the timing adjustment techniques on cellular-based OFDMA systems, the compromised timing adjustment (not the exact timing offset compensation) should be performed at the receiver in order to alleviate the MNI on OFDMA mesh network. By exploiting analyzed SINR expression, we proposed the new timing adjustment criterion to maximize the performance on mesh network without big burden on complexity and feedback. Simulation results verify the proposed timing adjustment techniques at the receiver outperform the exact timing compensation techniques, and the performance improvement by the proposed adjustment is more significant as the system is more crowd or the network size grows.

3.6 Appendix

If the timing adjustment θ_{pq} is given at the node rx q, it is determined to calculate the interference from the current symbol (CS), previous symbol (PS), and next symbol (NS), respectively. Based on [18], we unified the variables and refined the functions in order to express MNI for all cases in OFDMA mesh network. Let us define $n_{ab,L}(l)$ and $n_{ab,U}(l)$ as follows:

$$\begin{aligned} n_{ab,L}(l) &= \theta_{ab} + (\mu_a - \mu_q) + l - N_g \\ n_{ab,U}(l) &= \theta_{ab} + (\mu_a - \mu_q) + l + N - 1. \end{aligned}$$

Then, the lower and upper sample bounds of CS within FFT window is described as $\underline{n_{ab,L}}(l) = \max(0, n_{ab,L}(l))$ and $\overline{n_{ab,U}}(l) = \min(N-1, n_{ab,U}(l))$. Then, the amount of node interference caused by CS, PS and NS can be given by

$$\Gamma_{pq,\mathrm{CS}}[k] = \sum_{m \in \mathcal{S}_{pq}} \sum_{l=0}^{L-1} \left| \gamma_{mk} \left(\underline{n_{pq,L}}(l), \overline{n_{pq,U}}(l) \right) \right|^2 \tag{3.7}$$

$$\Gamma_{pq,\mathrm{PS}}[k] = \sum_{m \in \mathcal{S}_{pq}} \sum_{l=0}^{L-1} \left| \gamma_{mk} \left(0, n_{pq,L}(l) - 1 \right) \right|^2 \tag{3.8}$$

$$\Gamma_{pq,\mathrm{NS}}[k] = \sum_{m \in \mathcal{S}_{pq}} \sum_{l=0}^{L-1} \left| \gamma_{mk} \left(n_{pq,U}(l) + 1, N - 1 \right) \right|^2 . \tag{3.9}$$

The normalized total interference on link $\mathcal{L}_{pq}$ is denoted as

$$\Gamma_{pq,\mathrm{T}}[k] = \Gamma_{pq,\mathrm{CS}}[k] + \Gamma_{pq,\mathrm{PS}}[k] + \Gamma_{pq,\mathrm{NS}}[k] \tag{3.10}$$

Then, the explicit analysis for the variances are given by

$$E_q^2[k] = \mathcal{E}_{pq} \sum_{l=0}^{L-1} \left| \gamma_{kk} \left(\underline{n_{pq,L}}(l), \overline{n_{pq,U}}(l) \right) \right|^2 \tag{3.11}$$

$$\sigma_{q,\mathrm{SSNI}}^2[k] = \mathcal{E}_{pq} \times \Gamma_{pq,\mathrm{T}}[k] - E_q^2[k] \tag{3.12}$$

$$\sigma_{q,\mathrm{SONI}}^2[k] = \sum_{q' \neq q} \mathcal{E}_{pq'} \times \Gamma_{pq',\mathrm{T}}[k] \tag{3.13}$$

$$\sigma_{q,\mathrm{OSNI}}^2[k] = \sum_{p' \neq p} \mathcal{E}_{p'q} \times \Gamma_{p'q,\mathrm{T}}[k] \tag{3.14}$$

$$\sigma_{q,\mathrm{OONI}}^2[k] = \sum_{p' \neq p} \sum_{q' \neq q} \mathcal{E}_{p'q'} \times \Gamma_{p'q',\mathrm{T}}[k]. \tag{3.15}$$

Chapter 4

Carrier Frequency Offset Estimation and Compensation for OFDMA Uplink Systems

Carrier frequency offsets (CFO) at the uplink of Orthogonal Frequency Division Multiple Access (OFDMA) destroy the orthogonality among the users and introduce Multi-User Interference (MUI) which severely degrades the performance of the system. We propose pilot based CFO estimation and channel estimation for the OFDMA Uplink. A null sub-carrier (zero pilot) based cost function is used to estimate the CFO of each user. The MUI for each user is minimized by optimizing the position of the null sub-carriers. The effect of the null sub-carrier position on the acquisition range and the identifiability of CFO estimation is analyzed as well. Sub-band hopping is proposed to achieve channel independent performance of the CFO estimation and to obtain channel estimates of each user. We prove that the identifiability of the CFO estimation is guaranteed in the presence of channel nulls and the acquisition range supported by the algorithm. In order to compensate the effect of CFO, decision directed (DD) MUI reconstruction and cancelation is used. The resulting algorithm can estimate and compensate both the integer and the fractional CFOs. The performance of the algorithms is evaluated through simulations and compared with existing approaches.

4.1 Introduction

Orthogonal Frequency Division Multiple Access (OFDMA) is an efficient multi-user multi-carrier technique which has been proposed for current and next generation wireless communication systems, for example IEEE 802.16e (mobile WiMAX) [20]. However, carrier frequency offsets (CFO) between the base-station (BS) and the mobile users (MU) arising due to local oscillator mismatch and/or the mobility of the users destroy the orthogonality among the user's sub-carriers. Inter carrier interference

(ICI) and multi-user interference (MUI) arising due to these CFOs severely degrade the performance of the whole system. Therefore, CFO estimation and compensation is imperative at the BS and the MUs.

Due to time delays and doppler spread caused by the mobility of users, CFO estimation and compensation only at the downlink of OFDMA systems is not sufficient and calls for CFO estimation at the uplink as well. However, CFO estimation is much more challenging at the uplink due to the presence of multiple CFOs and the fact that the received signal at the BS is the sum of the transmitted signals from all the users. In addition, ensuring the identifiability of the CFO estimates is also an important requirement. By investigating the identifiability, we can determine whether a unique estimate for CFO exists or not. Ensuring identifiability in OFDMA is much more critical than OFDM [21, 22] as apart from channel nulls, MUI resulting from the CFOs of the other users can disturb the uniqueness of the CFO estimate.

Various solutions for CFO estimation and compensation for OFDMA have been proposed in the literature, but only few of them also discuss the identifiability. These solutions can be categorized on the basis of sub-carrier allocation scheme for OFDMA like sub-band allocation [1], [23], interleaved allocation [24] and the generalized allocation [25–27] or, on the basis of data aided [27] or non data aided [23] CFO estimation. These algorithms vary in their performance, complexity and acquisition range. In this report, we deal with the CFO estimation for OFDMA uplink and its identifiability. We use a null sub-carrier (zero pilot) based cost function to estimate the CFO of each user. For the sake of simplicity we deal with sub-band allocation, however the mathematical development will show that the proposed scheme can be easily extended to generalized sub-carrier allocation, which is being used in next generation wireless communication systems like IEEE 802.16m [20], 3GPP LTE-Advanced.

The other main contribution is the detailed discussion on the identifiability of CFO estimation and the effects of MUI in causing ambiguity in CFO estimation. We analyze the effect of changing the position of null sub-carriers on the acquisition range and the identifiability of CFO estimation. We show that placing the null sub-carriers in the middle of the sub-band not only minimizes the MUI but also maximizes the acquisition range. We prove that the identifiability of the CFO estimation is guaranteed for the acquisition range supported by the algorithm. In addition, we also discuss the effect of channel nulls on the identifiability and propose hopping technique to guarantee the identifiability in the presence of channel nulls.

The presence of multiple CFOs at the uplink also makes the CFO compensation a non-trivial task, because compensating the effect of CFO of one user further mis-aligns the other users and increases the MUI. Thus, a straight forward CFO compensation similar to OFDM cannot be employed for OFDMA. Various CFO compensation algorithms have been proposed in the literature like [2, 28–30]. Successive parallel interference cancelation is used in [30] to compensate the CFO of the users one by one. [28] uses an interference cancelation scheme based on circular convolution while a linear de-correlator detector is used in [2] to jointly compensate all the users. However, the CFO compensation in [2] has two disadvantages. It can only be applied for CFO less than half of the sub-carrier spacing and is computationally complex because it requires the inversion of a large matrix. In contrast to these existing approaches, we use a decision-directed (DD) MUI cancelation scheme to compensate

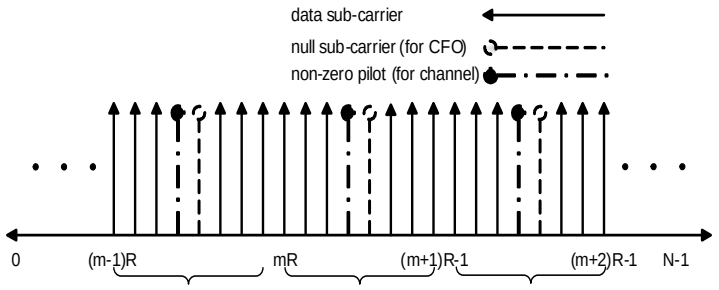


Figure 4.1: Sub-band carrier allocation for different users showing the null sub-carriers and channel estimation pilots within the sub-band, $R = 8$, $K = 1$, $P = 1$.

the effect of CFO of each user. We show that our CFO compensation scheme is independent of the acquisition range in contrast to [2] and [29] which fail to compensate integer offsets. We use pilot sub-carriers to estimate the channel of each user. These pilots are further employed to decrease the residual CFO error and an iterative scheme to further improve the CFO estimation has been proposed.

Notation. The subscript $(\cdot)_m$ is used to denote the m^{th} user. Upper (lower) boldface letters represent matrices (column vectors). $(\cdot)^H$ represents Hermition, $(\cdot)^T$ represents transpose and $(\cdot)^*$ represents conjugate. $D_N(\boldsymbol{h})$ with a vector argument represents an $N \times N$ diagonal matrix with vector $\boldsymbol{h}$ containing the diagonal entries while $D_N(f)$ with a scalar argument represents an $N \times N$ diagonal matrix with $[D_N(f)]_{nn} = e^{j\frac{2\pi}{N}fn}$. $\mathbf{I}_N$ represents an $N \times N$ identity matrix. We define $\mathbf{f}_N(w)$ as the $N \times 1$ normalized Fourier vector given as $\mathbf{f}_N(w) = N^{-1/2}[1, e^{jw}, e^{2jw}, ..., e^{j(N-1)w}]^T$.

4.2 System Model

Consider a base-band equivalent model of OFDMA uplink system. The system has N sub-carriers and can accommodate $M_{\max}$ users at maximum. The number of active users is M. As sub-band allocation scheme is used, m^{th} user is allocated a sub-band with R_m number of sub-carriers. For simplicity, we take $R_m = R = \lfloor \frac{N}{M} \rfloor$ so that the number of sub-carriers in each sub-band is equal. Each user employs K null sub-carriers to estimate its CFO. The number of non-zero sub-carriers allocated to the m^{th} user is therefore, $\tilde{R} = R - K$. These sub-carriers contain D data sub-carriers as well as P non-zero pilots used for channel estimation. Let I_{mn}, I_{mp} and I_{md}, be the sets of indices of null sub-carriers (zero pilots), non-zero pilots and data sub-carriers allocated to the m^{th} user, respectively. Then $I_{mn} := \{mR + n_k | n_k \in [0, 1, ..., R-1]\}$, where k = 1,...,K, $I_{mp} := \{mR + n_p | n_p \in [0, 1, ..., R-1]\}$, where p = 1,...,P, $I_{mn} \bigcap I_{mp} = \phi$ and $I_{md} := \{mR + d | d \in [0, 1, ...R-1] \wedge d \neq n_k \, d \neq n_p\}$,where d = 1,...,D. Thus $R_m = R = D + P + K$ and $I_{md} \bigcup I_{mp} \bigcup I_{mn} = \{mR, mR+1, ..., (m+1)R-1\}$. For ease of discussion, we assume that the position of pilots within the sub-band is the same for all the users, i.e., n_k and n_p are independent of m. Figure 4.1 shows an example of sub-band allocation with R = 8, $K = 1$ and $P = 1$.

Each user maps its data points on to its allocated sub-carriers. The q^{th} OFDMA symbol transmitted by the m^{th} user, before inverse discrete Fourier transform (IDFT), is given as

$$\mathbf{s}_m(q) = \mathbf{T}_m \mathbf{x}_m(q), \tag{4.1}$$

where $\mathbf{x}_m(q)$ is an $\tilde{R} \times 1$ vector which contains the information data and pilots for channel estimation, for the q^{th} symbol of the m^{th} user. $\mathbf{T}_m$ is the $N \times \tilde{R}$ sub-carrier allocation matrix for the m^{th} user.

$$\mathbf{T}_m = [e_{x_0}, e_{x_1}, ..., e_{x_{\tilde{R}-1}}],$$

where e_x is the x^{th} column of $\mathbf{I}_N$ and x_i is the i^{th} element of $I_{md} \bigcup I_{mp}$. $\mathbf{x}_m(q)$ is defined as,

$$\mathbf{x}_m(q) = \mathbf{T}_{md}\mathbf{d}_m(q) + \mathbf{T}_{mp}\mathbf{p}_m(q), \tag{4.2}$$

where $\mathbf{d}_m(q)$ and $\mathbf{p}_m(q)$ are the $D \times 1$ and $P \times 1$ vectors which contain the data points and pilot symbols of the m^{th} user respectively. $\mathbf{T}_{md}$ and $\mathbf{T}_{mp}$ are the corresponding $\tilde{R} \times D$ and $\tilde{R} \times P$ mapping matrices which map the data points and pilots symbols on their corresponding positions within the sub-band. Note that $\mathbf{T}_{md}$ and $\mathbf{T}_{mp}$ are mutually orthogonal and can be defined in a similar way as $\mathbf{T}_m$.

We assume that the base-band equivalent channel impulse response $\tilde{h}_m(l), l \in [0, L_m]$ of the m^{th} user is of order L_m. Therefore, a cyclic prefix (CP) of N_{CP} samples is appended at the start of each OFDMA symbol to mitigate inter-symbol interference (ISI) caused by the multi-path channel, where $N_{CP} \geq \max_m\{L_m\}$. Thus, the total number of samples in the transmitted symbol is $N_t = N + N_{CP}$. The q^{th} transmitted symbol after IDFT and CP insertion is

$$\hat{\mathbf{s}}_m(q) = \mathbf{T}_{cp}\mathbf{F}^H\mathbf{s}_m(q),$$

where $\mathbf{F}$ is an $N \times N$ unitary DFT matrix. $\mathbf{T}_{cp}$ is the $N_t \times N$ CP insertion matrix where $\mathbf{T}_{cp} = \begin{bmatrix} \mathbf{0}_{N_{CP}\times(N-N_{CP})} & \mathbf{I}_{N_{CP}} \\ & \mathbf{I}_N \end{bmatrix}$.

Assuming all the users are synchronized in time, the received signal for the m^{th} user at the BS is the sum of the signals transmitted by all the M active users and can be expressed as,

$$\begin{aligned}\mathbf{y}(q) &= \sum_{m=0}^{M-1} \mathbf{y}_m(q) \\ &= \sum_{m=0}^{M-1} e^{j\frac{2\pi}{N}f_m(qN_t+N_{CP})}\mathbf{D}_N(f_m)\mathbf{F}^H\mathbf{D}_N(\tilde{\mathbf{h}}_m)\mathbf{s}_m(q) + \mathbf{v}(q),\end{aligned}$$

where $f_m = \tilde{f}_m T$ is the normalized CFO of the m^{th} user and T is the sampling period. Note that $f_m \leq |f_{\max}|$ where $f_{\max}$ is the maximum allowable CFO. $\tilde{\mathbf{h}}_m$ is the $N \times 1$ vector containing the N point DFT of the m^{th} user's channel impulse response. $\mathbf{v}(q)$ is the $N \times 1$ vector containing the additive white Gaussian noise (AWGN) samples for the q^{th} symbol. Taking a DFT of the received signal at the BS results in

$$\begin{aligned}\mathbf{Y}(q) &= \mathbf{F}\mathbf{y}(q) \\ &= \sum_{m=0}^{M-1} e^{j\frac{2\pi}{N}f_m(qN_t+N_{CP})}\mathbf{F}\mathbf{D}_N(f_m)\mathbf{F}^H\mathbf{T}_m\mathbf{D}_{\tilde{R}}(\mathbf{h}_m)\mathbf{x}_m(q) + \mathbf{F}\mathbf{v}(q).\end{aligned}$$

where $\mathbf{h}_m = [\tilde{h}_m(i)|i \in I_{md} \bigcup I_{mp}]$, $i = 1, ..., \tilde{R}$ is a $\tilde{R} \times 1$ vector containing the channel frequency response at the non-zero sub-carriers of the m^{th} user. Note that if $f_m = 0$ for each user, $\mathbf{F}\mathbf{D}_N(f_m)\mathbf{F}^H = \mathbf{I}_N$, and the users are perfectly separable and channel estimation can be performed straightaway. However, if f_m is non zero, $\mathbf{F}\mathbf{D}_N(f_m)\mathbf{F}^H$ gives a non-diagonal matrix, which leads to ICI as well as MUI. Therefore, estimation and compensation of the CFO of each user is necessary for channel estimation and decoding the signals transmitted from each user.

4.3 CFO Estimation Algorithm

The block diagram of CFO estimation algorithm is shown in the Figure 4.2. We use minimization of a null sub-carrier based cost function to estimate the CFO of each user, which is an extension of [31]. We observe the interference over the null sub-carriers of each user and minimize it to estimate the CFO. The result is a cost function which uses a line search to estimate the CFO of each user separately. From here onwards, we use m' to indicate the user for which the CFO estimation is being carried out, and m to indicate the other users. The autocorrelation matrix of the received symbol, assuming the transmitted symbols are uncorrelated is given as,

$$\begin{aligned}\mathbf{R}_{\mathbf{yy}} &= E\{\mathbf{y}\mathbf{y}^H\} \\ &= \sum_{m=0}^{M-1} \mathbf{D}_N(f_m)\mathbf{F}^H\mathbf{T}_m\mathbf{D}_{\tilde{R}}(\mathbf{h}_m)\mathbf{R}_{\mathbf{x}_m\mathbf{x}_m}\mathbf{D}_{\tilde{R}}^H(\mathbf{h}_m)\mathbf{T}_m^H\mathbf{F}\mathbf{D}_N^H(f_m) + \sigma_v^2\mathbf{I}_N,\end{aligned} \tag{4.3}$$

where $\mathbf{R}_{\mathbf{x}_m\mathbf{x}_m}$ and $\sigma_v^2\mathbf{I}_N$ are the autocorrelation matrices of $\mathbf{x}_m(q)$ and $\mathbf{v}(q)$ respectively. Note that in the absence of noise, the Fourier vector corresponding to each null sub-carrier of the m'^{th} user is orthogonal to the autocorrelation matrix when $f_m = 0$ for each m, i.e., $\mathbf{f}_N^H(\frac{2\pi}{N}\{m'R + n_k\})\mathbf{R}_{\mathbf{yy}} = \mathbf{0}_{1\times N}$ for each n_k when $f_m = 0$. Thus a cost function for the m'^{th} user employing its null sub-carriers can be formed as,

$$J_{m'}(f_o) = \sum_{k=1}^{K} \mathbf{f}_N^H(\frac{2\pi}{N}\{m'R + n_k\})\mathbf{D}_N^{-1}(f_o)\mathbf{R}_{\mathbf{yy}}\mathbf{D}_N(f_o)\mathbf{f}_N(\frac{2\pi}{N}\{m'R + n_k\}), \tag{4.4}$$

where $f_o \leq |f_{\max}|$ is the candidate CFO estimate for the m'^{th} user. Such a cost function is used for each user employing its own null sub-carriers only and the minimum of the cost function is taken as the estimate of the CFO of the m'^{th} user;

$$\hat{f}_{m'} = \arg\min_{f_o} \left(J_{m'}(f_o)\right).$$

In practical implementation, $\mathbf{R}_{\mathbf{yy}}$ is estimated through its sample estimate

$$\hat{\mathbf{R}}_{\mathbf{yy}} = \frac{1}{Q}\sum_{q=1}^{Q} \mathbf{y}(q)\mathbf{y}^H(q), \tag{4.5}$$

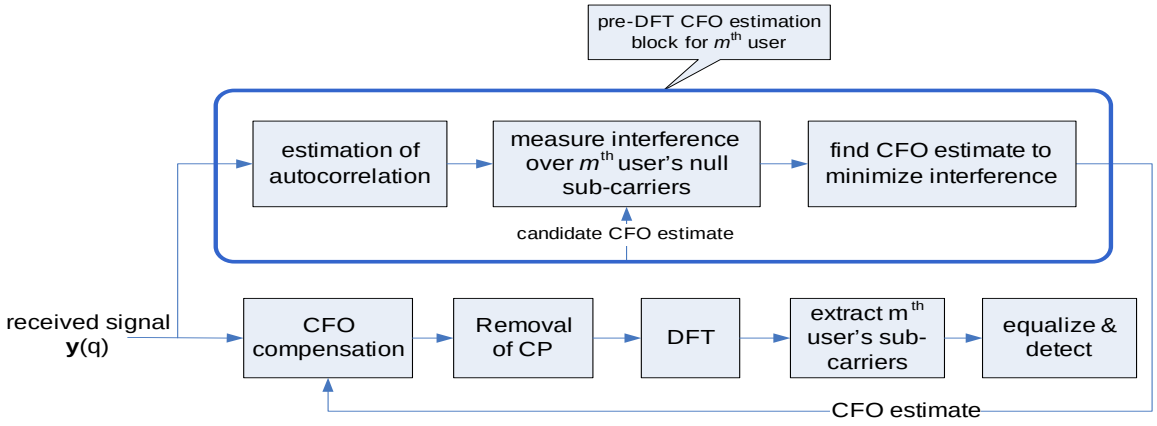


Figure 4.2: Block diagram of CFO estimation algorithm

and provided Q is sufficiently large, (4.3) is also satisfied for $\hat{\mathbf{R}}_{\mathbf{yy}}$ with $\mathbf{R}_{\mathbf{x}_m\mathbf{x}_m}$ replaced by $\hat{\mathbf{R}}_{\mathbf{x}_m\mathbf{x}_m}$ and the analysis and conclusions drawn for $\mathbf{R}_{\mathbf{yy}}$ are also applicable to $\hat{\mathbf{R}}_{\mathbf{yy}}$. The number of symbols sufficient to estimate $\mathbf{R}_{\mathbf{yy}}$ is discussed in Section 4.4.4. Using (4.3), the cost function in (4.4) can be written as a sum of an ICI term which we call the "self interference" (SI) term and the MUI term as follows,

$$J_{m'}(f_o) = J_{m',\mathrm{SI}}(f_o) + J_{m',\mathrm{MUI}}(f_o), \tag{4.6}$$

where

$$\begin{aligned} J_{m',\mathrm{SI}}(f_o) = & \sum_{k=1}^{K} \mathbf{f}_N^H \left(\frac{2\pi}{N}\{m'R + n_k - f_{m'} + f_o\} \right) \mathbf{F}^H \mathbf{T}_{m'} \mathbf{D}_{\tilde{R}}(\mathbf{h}_{m'}) \mathbf{R}_{\mathbf{x}_{m'}\mathbf{x}_{m'}} \mathbf{D}_{\tilde{R}}^H(\mathbf{h}_{m'}) \mathbf{T}_{m'}^H \mathbf{F} \\ & \mathbf{f}_N \left(\frac{2\pi}{N}\{m'R + n_k - f_{m'} + f_o\} \right) + \sigma_v^2, \end{aligned}$$

and

$$J_{m',\mathrm{MUI}}(f_o) = \sum_{m=0, m\neq m'}^{M-1} \tilde{J}_m(f_o),$$

with

$$\begin{aligned} \tilde{J}_m(f_o) = & \sum_{k=1}^{K} \mathbf{f}_N^H \left(\frac{2\pi}{N}\{m'R + n_k - f_m + f_o\} \right) \mathbf{F}^H \mathbf{T}_m \mathbf{D}_{\tilde{R}}(\mathbf{h}_m) \mathbf{R}_{\mathbf{x}_m\mathbf{x}_m} \mathbf{D}_{\tilde{R}}^H(\mathbf{h}_m) \mathbf{T}_m^H \mathbf{F} \\ & \mathbf{f}_N \left(\frac{2\pi}{N}\{m'R + n_k - f_m + f_o\} \right) + \sigma_v^2. \end{aligned}$$

As the noise term is independent of the CFO, it can be dropped from the cost function. Note that when $f_o = f_{m'}$ in (4.6), the SI will be completely eliminated whereas the MUI can still be non-zero (unless all the CFOs are the same). The accuracy of CFO estimation is affected by the MUI on the null sub-carriers used for CFO estimation. If we can minimize the MUI for the null sub-carriers, the effect of this MUI on CFO estimation will be minimized. Therefore, the main idea behind our CFO estimation is that we find the CFO point to eliminate the SI through line search and simultaneously

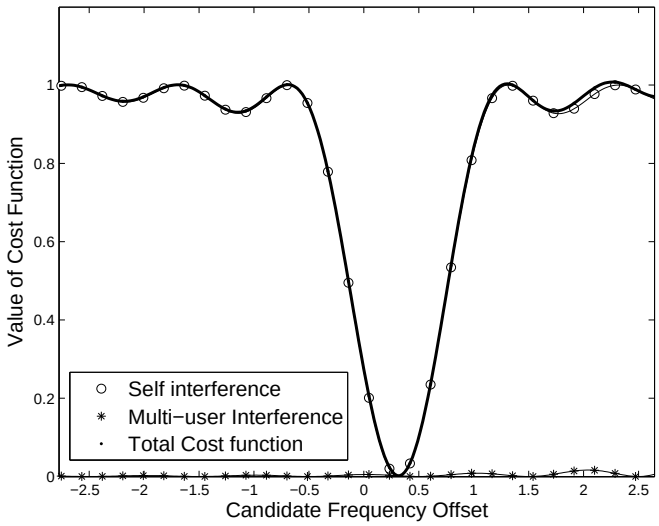


Figure 4.3: A typical plot showing the SI , MUI term and the total CFO estimation cost function

minimize the MUI by optimizing the location of null sub-carriers within the sub-band, so that the minimum of the total cost function is the same as the minimum of the SI term. Figure 4.3 shows a typical plot for the SI and MUI terms along with the total cost function.

After some algebraic manipulations, the SI and MUI terms in (4.6) can be rewritten as,

$$J_{m',\mathrm{SI}}(f_o) = \frac{1}{N^2}\sum_{k=1}^{K}\sum_{r=0,r\neq n_k}^{R-1}\sigma^2_{\mathbf{x}_{m'}}|\tilde{h}_{m'}(m'R+r)|^2\psi^2_{\Delta m',\Delta r}(f_{m'}-f_o),$$

and

$$J_{m',\mathrm{MUI}}(f_o) = \sum_{m=0,m\neq m'}^{M-1}\tilde{J}_m(f_o),$$

with

$$\tilde{J}_m(f_o) = \frac{1}{N^2}\sum_{k=1}^{K}\sum_{r=0,r\neq n_k}^{R-1}\sigma^2_{\mathbf{x}_m}|\tilde{h}_m(mR+r)|^2\psi^2_{\Delta m,\Delta r}(f_m-f_o), \tag{4.7}$$

where $\psi_{\Delta m,\Delta r}(x) = \frac{\sin(\pi x)}{\sin\left(\frac{\pi}{N}(x+R\Delta m+\Delta r)\right)}$is the *normalized interference* function, $\Delta m = m - m'$ and $\Delta r = r - n_k$. It represents the normalized interference caused by the r^{th} sub-carrier of the m^{th} user on the null sub-carrier (n_k) of the m'^{th} user due to f_m, evaluated at the candidate CFO, i.e., f_o. $\psi_{\Delta m,\Delta r}(x)$ is a sinc type function whose magnitude is maximum in the center of its main lobe. Δm represents the distance of the m^{th} user from the main lobe in units of the sub-band size R with $\Delta m'$ $= 0$, and Δr represents the distance of the r^{th} sub-carrier of the m^{th} user from the null sub-carrier n_k with in the sub-band. The magnitude of $\psi^2_{\Delta m,\Delta r}(x)$ decreases as $|\Delta m|$ and $|\Delta r|$ increase. Therefore in (4.7), the normalized MUI terms for $|\Delta m| = \pm 1$, corresponding to the users $(m' \pm 1)$, are dominant and

the normalized MUI for these adjacent interfering users is maximum when the main lobe of $\psi_{\Delta m,\Delta r}(x)$ is sampled. Eq. (4.7) also suggests that for a given f_m and f_o, the interference caused by the $(m'+1)^{th}$ user on the null sub-carrier n_k is greater than the $(m'-1)^{th}$ user if the null sub-carrier is closer to the $(m'+1)^{th}$ user and vice versa. This suggests that the MUI is minimum if the null sub-carrier is placed at the center of the sub-band.

4.4 Identifiability and acquisition range of CFO estimation

Ensuring identifiability means that it is guaranteed for the cost function to contain a unique minimum. As the cost function contains the SI as well as the MUI, the MUI can cause the cost function to have multiple minimum and thus creating confusion in the CFO estimation. Moreover, the channel nulls at the sub-carriers of the users can also cause multiple minimum in the cost function. In this section, we analyze the effect of MUI, null sub-carrier position and the channel on the cost function and determine its acquisition range. Then we prove that for the given acquisition range, the cost function always has a unique minimum.

4.4.1 Effect of SI on identifiability and acquisition range

At first, consider the effect of the SI term and ignore the MUI. Note that, the SI term in (4.6) is zero if,

$$\mathbf{f}_N^H(\frac{2\pi}{N}\{m'R+n_k-f_{m'}+f_o\})\mathbf{F}^H\mathbf{T}_{m'}=\mathbf{0}, \tag{4.8}$$

for each of the null sub-carrier, i.e., $\mathbf{f}_N^H(\frac{2\pi}{N}\{m'R+n_k-f_{m'}+f_o\})$ should lie in the left null space (LNS) of $\mathbf{F}^H\mathbf{T}_{m'}$. As $\mathbf{F}^H\mathbf{T}_{m'}$ only contains the Fourier vectors corresponding to the non-zero sub-carriers of the m'^{th} user, its LNS contains the Fourier vectors corresponding to two sub-carrier groups; the null sub-carriers of the m'^{th} user $(I_{m'n})$ as well as all the sub-carriers of all the other users $(I_{md}\bigcup I_{mp}\bigcup I_{mn}\forall m\neq m')$. Therefore, when $f_o=f_{m'}$, condition in (4.8) is satisfied as $(m'R+n_k-f_{m'}+f_o)\in I_{m'n}$ however as f_o is varied, (4.8) may again be satisfied if $(m'R+n_k-f_{m'}+f_o)\in I_{md}\bigcup I_{mp}\bigcup I_{mn}\forall m\neq m'$ which causes multiple minimum in the cost function. Therefore, CFO is identifiable if $(m'R+n_k-f_{m'}+f_o)$ is restricted to the sub-carriers of the m'^{th} user only, i.e,

$$m'R-1<m'R+n_k-f_{m'}+f_o<m'R+R,$$

$$n_k-R<f_{m'}-f_o<n_k+1. \tag{4.9}$$

Since $f_{m'}\leq|f_{\max}|$ as mentioned before, $f_{m'}-f_o\leq|2f_{\max}|$. Therefore in order to satisfy the condition in (4.9) the maximum allowable CFO should be,

$$f_{\max}=\frac{1}{2}\min_k\left(|n_k+1|,|n_k-R|\right), \tag{4.10}$$

when only the effect of SI is considered. If $f_{m'}>f_{\max}$, the condition in (4.9) will not be satisfied for some values of $f_{m'}$ and f_o, which means the CFO estimate will not be unique.

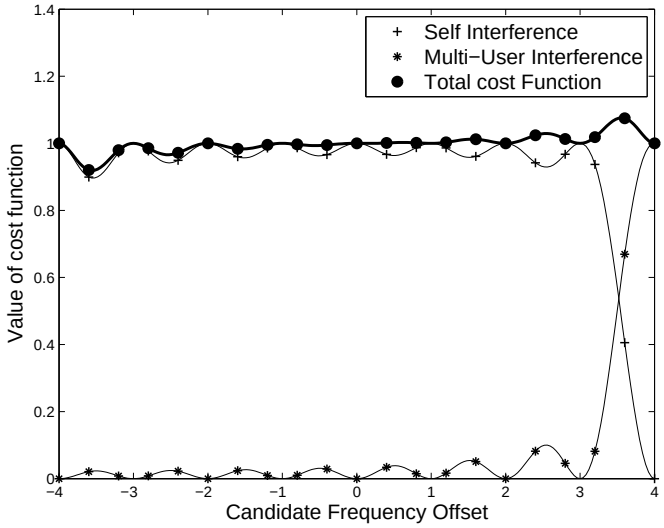


Figure 4.4: Ambiguity in CFO Estimation when $f_{m'} = \frac{R}{4}$ and $f_m = -\frac{R}{4}$, $R = 16$, $K = 1$, $n_k = \frac{R}{2}$.

4.4.2 Effect of MUI on identifiability and acquisition range

Now we consider the effect of MUI on the acquisition range and the identifiability. As (4.7) suggests that the MUI term is non-negative, it can raise the minimum of the cost function from zero to a non-zero point, and thus analyzing its behavior at the minimum of the SI term is important. The power of MUI depends on the null sub-carrier position as discussed in the previous section, but it also depends on the magnitude of $f_m - f_o$. As the candidate CFO, i.e., f_o is varied, the value of MUI also changes and for the worst case when $f_m - f_o = \pm 2 f_{max}$ for $\Delta m = \pm 1$, (4.6) suggests that the MUI for one of the adjacent users will attain its maximum value as $J_{m',\mathrm{MUI}}(f_o)$ will have a sample form the main lobe of $\psi^2_{\Delta m,\Delta r}(f_m - f_o)$. If $f_o = f_m$ is also satisfied at this point, the maximum of MUI will be exactly at the minimum of SI and the minimum of cost function will be vanished. This worst case MUI occurs when $f_{m'} = \pm f_{\max}$ and $f_{m'\pm 1} = -f_{m'}$. Figure 4.4 shows an example of such a worst case MUI. As shown, the minimum of SI at $f_{m'} = \frac{R}{4}$ gets vanished by MUI and the total cost function gives a wrong CFO estimate. Therefore, the maximum identifiable CFO range should be reduced from (4.10) to

$$f_{\max} = \frac{1}{2}\left(\min_k\left(|n_k + 1|, |n_k - R|\right) - 1\right), \tag{4.11}$$

when the effect of both SI and MUI is considered. Under this refined acquisition range, the adjacent interfering users can have samples from the first side lobe of $\psi^2_{\Delta m,\Delta r}(x)$ at most whereas the current user has samples from the main lobe of $\psi^2_{\Delta m',\Delta r}(x)$. We prove that the identifiability of CFO estimation is always guaranteed for $f_{\max}$ in (4.11), in the next section.

The acquisition range of CFO estimation is therefore, $\frac{\Delta f}{2}(\min_k(|n_k + 1|, |n_k - R|) - 1)$ where Δf is the sub-carrier spacing. This shows that the acquisition range is dependent on the null sub-carrier position. Maximum acquisition range is $\pm\frac{\Delta f}{2}\left(\frac{R}{2} - 1\right)$ which is achieved for $n_k = \frac{R}{2}$, i.e., when the null sub-carrier is at the middle of the sub-band. This shows that the null sub-carrier in the middle

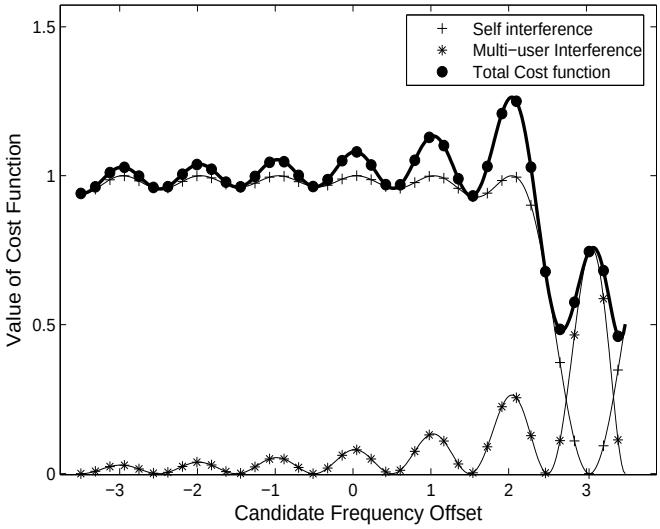


Figure 4.5: A hypothetical worst case MUI scenario causing ambiguity in CFO estimation at $f_{m'} = 3$.

not only results in lower MUI but also results in higher acquisition range.

4.4.3 Identifiability with refined acquisition range

Both, channel nulls and MUI can cause the cost function to have multiple minimum and thus creating confusion in CFO estimation. We deal with the identifiability issues caused by the channel in the next section and thus assume here that the channel matrix $\mathbf{D}_{\tilde{R}}(\mathbf{h}_m)$ is full rank.

MUI can cause identifiability issues in two ways. First case has been dealt with in the section 4.4.2, so that the magnitude of MUI is never too large to cause the minimum of the cost function to vanish. The other case, shown for purpose of illustration in Figure 4.5, occurs if the MUI adds to the SI and the total cost function has two minima. As shown, the MUI term has more sharp peak as compared to the SI term and raises the minimum of SI term at $f_o = 3$ to cause two minima in the cost function. As the curvature signifies the measure of peakedness of a curve [32], the case shown in Figure 4.5 can never occur if the curvature of the SI term at its minimum is greater than the curvature of the MUI term.

Theorem: If the maximum allowable CFO range is given by (4.11), the curvature of the normalized MUI term is always less than the curvature of the SI term at its minimum and thus, the identifiability of the CFO estimation is always guaranteed.

Proof: We assume that all the users have equal power and thus deal with normalized interference. Note that the curvature of the MUI term at positions other than the minimum of the SI term does not cause any identifiability problem. Therefore, we model a worst case MUI scenario in which the MUI has maximum curvature at the minimum of the SI term. We then prove that the curvature of this worst case MUI is less than the curvature of the SI term at its minimum and thus the CFO estimate is unique. The normalized MUI $\mathcal{J}_{m',\mathrm{MUI}}(f_o)$ in (4.7) with all users active is,

$$\mathcal{J}_{m',\mathrm{MUI}}(f_o) = \sum_{m=0,m\neq m'}^{M_{\max}-1} \tilde{\mathcal{J}}_m(f_o),$$

where

$$\tilde{\mathcal{J}}_m(f_o) = \frac{1}{N^2} \sum_{r=0,r\neq n_k}^{R-1} \psi^2_{\Delta m,\Delta r}(f_m - f_o). \tag{4.12}$$

The curvature of the normalized MUI at any point is given as [32],

$$\kappa_{\mathcal{J}_{m',\mathrm{MUI}}}(f_o) = \frac{\left|\frac{\partial^2 \mathcal{J}_{m',\mathrm{MUI}}(f_o)}{\partial f_o^2}\right|}{\left(1 + \left(\frac{\partial \mathcal{J}_{m',\mathrm{MUI}}(f_o)}{\partial f_o}\right)^2\right)^{\frac{3}{2}}}.$$

Similarly the curvature of the SI term $\kappa_{\mathcal{J}_{m',\mathrm{SI}}}(f_o)$ can also be defined. The curvature of each $\tilde{\mathcal{J}}_m(f_o)$ term in (4.12) is maximum when an extreme point of a side lobe of $\psi^2_{\Delta m,\Delta r}(f_m - f_o)$ is sampled. According to (4.12), this happens whenever $f_m - f_o = i$ or $f_m - f_o = i \pm \frac{1}{2}$ where i is an integer. We are interested in the case when $f_m - f_0 = i \pm \frac{1}{2}$ because in that case the maximum of the side lobe is sampled. At a maximum point, larger magnitude of second derivative implies larger curvature. Also, in order for MUI to have maximum curvature, each interfering user must have the same f_m because in such a case, all the $\tilde{\mathcal{J}}_m(f_o)$ terms will be coherent, and MUI will have maximum curvature at $f_m - f_o = \pm(2f_{\max} - \frac{1}{2})$. So,

$$\max\left(\kappa_{\mathcal{J}_{m',\mathrm{MUI}}}(f_o)\right) = \kappa_{\mathcal{J}_{m',\mathrm{MUI}}}(f_o)|_{(f_m=f,\forall m\neq m')},$$

at $f_m - f_o = \pm(2f_{\max} - \frac{1}{2})$, where f is a constant frequency offset.

Also, in order to destroy uniqueness of the CFO estimate, this maximum curvature of the MUI term should occur at the minimum of the SI term. This condition is satisfied in two cases. One, when $f_m = f_{\max}$ for all the interfering users and $f_{m'} = -f_{\max} + \frac{1}{2}$ for $n_k < \frac{R}{2}$, and second, when $f_m = -f_{\max}$ for all the interfering users and $f_{m'} = f_{\max} - \frac{1}{2}$ for $n_k \geq \frac{R}{2}$. In both of these cases, minimum of the SI term occurs at $f_o = f_{m'}$ and at the same point $f_m - f_o = \pm(2f_{\max} - \frac{1}{2})$ is also satisfied and the worst case MUI term has maximum curvature at the minimum of the SI term. We prove the identifiability for this worst case scenario by showing that

$$\kappa_{\mathcal{J}_{m',\mathrm{SI}}}(f_o) > \max\left(\kappa_{\mathcal{J}_{m',\mathrm{MUI}}}(f_o)\right),$$

at $f_o = f_{m'}$, in the Appendix.

4.4.4 Effect of channel on identifiability

Identifiability of CFO estimation is also affected by the channel response at the DFT grid corresponding to the data sub-carriers allocated to the user in consideration. In the previous sections, we ignored the effect of channel by assuming that the channel matrix $\mathbf{D}_{\tilde{R}}(\mathbf{h}_m)$ is full rank, i.e., there is no null at

the channel response corresponding to the non-zero sub-carriers of the m'^{th} user. In this section, we consider the effect of these channel nulls on the identifiability of CFO estimation.

In (4.6), the SI term is zero if

$$\sum_{k=1}^{K} \mathbf{f}_N^H(\frac{2\pi}{N}\{m'R + n_k - f_m + f_o\})\mathbf{F}^H\mathbf{T}_{m'}\mathbf{D}_{\tilde{R}}(\mathbf{h}_{m'}) = 0, \tag{4.13}$$

for each n_k. Now, let us the suppose that the channel for the m'^{th} user has I nulls at $\{\gamma_{m',i}\}_{i=1}^{I}$ sub-carriers where each $\gamma_{m',i} \subset I_{m'd} \bigcup I_{m'p}$. Then the above condition is satisfied if

$$m'R + n_k - f_{m'} + f_o \subset \{\gamma_{m',i}\}_{i=1}^{I} \bigcup I_{m'n},$$

for each k. This condition is clearly satisfied for $f_o = f_{m'}$. But it can also be satisfied for other cases as well. Let us first consider the case, when $K = 1$. As

$$-2f_{\max} \leq -f_{m'} + f_o \leq 2f_{\max},$$

a channel null satisfying

$$n_k - 2f_{\max} \leq \gamma_{m',i} \leq n_k + 2f_{\max},$$

can cause (4.13) to be zero for $f_o \neq f_{m'}$. The second case is in which each user contains consecutive null sub-carriers. In that case (4.13) is satisfied if $\{\gamma_{m',i}\}_{i=1}^{I} \bigcup I_{m'n}$ has K consecutive elements not all taken from the set $I_{m'n}$ or if $\{\gamma_{m',i}\}_{i=1}^{I}$ has K consecutive elements. In either case (4.13) is satisfied for $f_o \neq f_{m'}$ causing multiple minima in the SI term, and consequently in the cost function. Thus one has to find solution in order to cope with this identifiability issue. It is important to note here that the channel response for a particular user at the sub-carriers other than its non-zero sub-carriers does not matter. We use sub-band hopping as an extension to null sub-carrier hopping in [1, 21] in order to avoid ambiguity in CFO identifiability due to channel nulls and to achieve channel independent performance.

Null sub-carrier hopping has been suggested in [21] to achieve channel independent performance in OFDM. It makes the sub-carrier allocation matrix $\mathbf{T}_m$ dependent on the OFDMA symbol index q. But, hopping in OFDMA is different from [21] because null sub-carrier hopping is not achieved by hopping the location of the null sub-carrier within the sub-band; instead we keep the null sub-carrier indices n_k, same in each sub-band and hop the sub-band of each user. Sub-band hopping is also used for our channel estimation algorithm as discussed in the next section. Specifically, we use a hopping index $n_q = q\lceil \frac{N}{L_{max}+1} \rceil$ dependent on the symbol index q which determines the circular shift in the start of the sub-band of each user, where $L_{max} = \max_m(L_m)$. Note that all the users must hop their sub-bands by the same amount n_q, because the BS receives the sum of transmitted signals from each user and cannot un-hop them individually. So, elements in $I_{md}(q)$, $I_{mp}(q)$ and $I_{mn}(q)$ become symbol dependent and now contain elements $mR + d + n_q$, $mR + n_p + n_q$ and $mR + n_k + n_q$ respectively. The received signal now becomes

$$\mathbf{y}(q) = \sum_{m=0}^{M-1} e^{j\frac{2\pi}{N} f_m(qN_t+N_{CP})} \mathbf{D}_N(f_m)\mathbf{F}^H\mathbf{T}_m(q)\mathbf{D}_{\tilde{R}}(\mathbf{h}_m(q))\mathbf{x}^m(q),$$

where $\mathbf{D}_{\tilde{R}}(\mathbf{h}_m(q))$ now contains the channel coefficients at the sub-carriers in $I_{md}(q) \bigcup I_{mp}(q)$, on its diagonal. In order to calculate $\mathbf{R}_{\mathbf{yy}}$, a un-hop matrix $\mathbf{D}_N^H(\frac{2\pi}{N}n_q)$ is applied to the received signal $\mathbf{y}(q)$. So, the un-hopped received signal becomes

$$\begin{aligned}\hat{\mathbf{y}}(q) &= \mathbf{D}_N^H(\frac{2\pi}{N}n_q)\mathbf{y}(q) \\ &= \sum_{m=0}^{M-1} e^{j\frac{2\pi}{N} f_m(qN_t+N_g)} \mathbf{D}_N(f_m)\mathbf{D}_N^H(\frac{2\pi}{N}n_q)\mathbf{F}^H\mathbf{T}_m(q)\mathbf{D}_{\tilde{R}}(\mathbf{H}_m(q))\mathbf{X}_m(q).\end{aligned}$$

Note that $\mathbf{D}_N^H(\frac{2\pi}{N}n_q)\mathbf{F}^H\mathbf{T}_m(q) = \mathbf{F}^H\mathbf{T}_m$ where $\mathbf{T}_m$ is same as in (4.6). The net effect of this hopping and un-hopping is that, the data sub-carriers for each OFDMA symbol of each user, are affected by a different channel response on the DFT grid. Calculating the autocorrelation matrix of un-hopped received signal, thus results in sort of averaging of channel response for each sub-carrier which results in channel independent CFO estimation. The autocorrelation matrix now becomes,

$$\mathbf{R}_{\hat{\mathbf{y}}\hat{\mathbf{y}}} = \sum_{m=0}^{M-1} \mathbf{D}_N(f_m)\mathbf{F}^H\mathbf{T}_m E\left[\mathbf{D}_{\tilde{R}}(\mathbf{H}_m(q))\mathbf{R}_{\mathbf{x}_m\mathbf{x}_m}\mathbf{D}_{\tilde{R}}(\mathbf{H}_m(q))\right]\mathbf{T}_m^H\mathbf{F}\mathbf{D}_N^H(f_m) + \sigma_v^2\mathbf{I}_N,$$

in which $E\left[\mathbf{D}_{\tilde{R}}(\mathbf{H}_m(q))\mathbf{R}_{\mathbf{x}_m\mathbf{x}_m}\mathbf{D}_{\tilde{R}}(\mathbf{H}_m(q))\right]$ should be full rank in order to have channel independent performance. We assume that the non-zero pilots and user's data symbols are uncorrelated from each other. Therefore, in order to have a full rank matrix $\mathbf{R}_{\mathbf{x}_m\mathbf{x}_m}$, the number of symbols Q in (4.5) used to estimate $\mathbf{R}_{\hat{\mathbf{y}}\hat{\mathbf{y}}}$ should be greater than M. The channel frequency response of each user can have at most L_m nulls on the DFT grid. Therefore, if $Q \geq \max(M, L_{\max}+1)$, $E[\mathbf{D}_{\tilde{R}}(\mathbf{h}_m(q))\mathbf{R}_{\mathbf{x}_m\mathbf{x}_m}\mathbf{D}_{\tilde{R}}(\mathbf{h}_m(q))]$ becomes full rank owing to the design of n_q.

4.5 CFO compensation and channel estimation

One of the key features of our CFO estimation algorithm is that the CFO of each user is estimated before performing the DFT at the BS. Therefore, CFO compensation for each user can also be applied before DFT.

4.5.1 CFO compensation and MUI cancelation

The block diagram of the CFO compensation, channel estimation and data detection in shown in Figure 4.6. The CFO of the m'^{th} user is compensated before DFT as

$$\hat{\mathbf{y}}_{m'}(q) = e^{-j\frac{2\pi}{N}\hat{f}_{m'}(qN_t+N_{CP})}\mathbf{D}_N^{-1}(\hat{f}_{m'})\mathbf{y}(q),$$

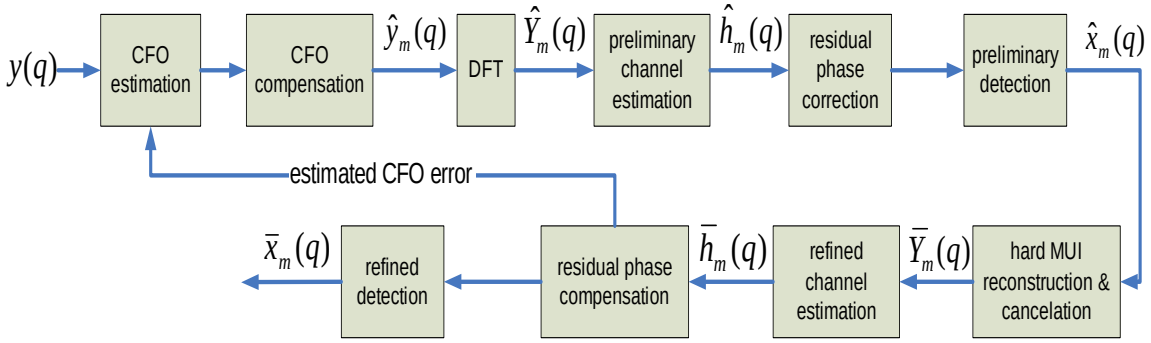


Figure 4.6: Block diagram of decision directed MUI compensation scheme

which is followed by the extraction of m'^{th} user's sub-carriers after performing DFT, to obtain preliminary estimate of m'^{th} user's received signal, i.e., $\hat{\mathbf{Y}}_m(q)$. However, the preliminary estimate of the received signal is plagued by the MUI arising due to the CFO of interfering users,

$$\begin{aligned}\hat{\mathbf{Y}}_{m'}(q) &= \mathbf{T}_{m'}^H \mathbf{F}\hat{\mathbf{y}}_{m'}(q) \\ &= e^{j\frac{2\pi}{N}(f_{m'}-\hat{f}_{m'})(qN_t+N_{CP})}\mathbf{T}_{m'}^H\mathbf{F}\mathbf{D}_N(f_{m'}-\hat{f}_{m'})\mathbf{F}^H\mathbf{D}_N(\mathbf{h}_{m'})\mathbf{T}_{m'}\mathbf{x}_{m'}(q)+\hat{\mathbf{v}}_{m'}(q) \\ &+ \sum_{m=0,m\neq m'}^{M-1} e^{j\frac{2\pi}{N}(f_m-\hat{f}_{m'})(qN_t+N_{CP})}\mathbf{T}_{m'}^H\mathbf{F}\mathbf{D}_N(f_m-\hat{f}_{m'})\mathbf{F}^H\mathbf{D}_N(\mathbf{h}_m)\mathbf{T}_m\mathbf{x}_m(q)+\hat{\mathbf{v}}_m(q).\end{aligned} \tag{4.14}$$

where $\hat{\mathbf{v}}_m(q) = e^{-j\frac{2\pi}{N}\hat{f}_m(qN_t+N_{CP})}\mathbf{D}_N^{-1}(\hat{f}_m)\mathbf{F}\mathbf{v}_m(q)$.

In the above equation, the first part corresponds to the m^{th} user's signal while the second part corresponds to the MUI. Note that if the CFO estimation is perfect, $\mathbf{D}_N(f_{m'}-\hat{f}_{m'}) = \mathbf{I}_N$ and ICI is completely eliminated but, $f_m - \hat{f}_{m'}$ is non-zero for each m which leads to non-zero MUI. The compensation of this MUI is necessary in order to obtain satisfactory channel estimation and detection performance. We analyze the effect of imperfect CFO estimation in section 4.5.3. We use a two stage DD MUI reconstruction scheme to cancel the effect of MUI. In the first stage, preliminary channel estimation and data detection is carried out for each user using $\hat{\mathbf{Y}}_m(q)$ and the channel estimation pilots to obtain $\hat{\mathbf{h}}_m(q)$ and $\hat{\mathbf{x}}_m(q)$. In the second stage, this preliminary channel and data information is used to obtain an estimate of MUI for each user. This MUI estimate is then subtracted from $\hat{\mathbf{Y}}_{m'}(q)$ to obtain the refined estimate of the received signal of each user, i.e., $\bar{\mathbf{Y}}_{m'}(q)$. Specifically,

$$\bar{\mathbf{Y}}_{m'}(q) = \hat{\mathbf{Y}}_{m'}(q) - \sum_{m=0,m\neq m'}^{M-1} e^{j\frac{2\pi}{N}(\hat{f}_m-\hat{f}_{m'})(qN_t+N_{CP})}\mathbf{T}_{m'}^H\mathbf{F}\mathbf{D}_N(\hat{f}_m-\hat{f}_{m'})\mathbf{F}^H\mathbf{T}_m\mathbf{D}_{\tilde{R}}(\hat{\mathbf{h}}_m(q))\hat{\mathbf{x}}_m(q),$$

$\bar{\mathbf{Y}}_{m'}(q)$ is then used to refine the channel estimates and to perform final data detection for each user. The main advantage of using this scheme is that it can be applied to acquisition ranges larger then than half of the sub-carrier spacing. In addition, we also propose a method to further improve the CFO estimation performance.

4.5.2 Channel Estimation

Least Square (LS) channel estimation is performed using the non-zero pilots inserted in the transmitted signal of each user in (4.2). At the BS, the pilot sub-carriers can be extracted from the compensated received signal of each user in order to perform channel estimation. Here, for the ease of discussion, we assume that we have perfect (MUI free) estimate $\mathbf{Y}_{m'}(q)$ of each user's received signal and use it to perform channel estimation. In practice, preliminary channel estimation is carried out using $\hat{\mathbf{Y}}_{m'}(q)$ and refined channel estimation uses $\bar{\mathbf{Y}}_{m'}(q)$ as discussed in section 4.5.1. The received pilot sub-carriers $\mathbf{z}_{m'}(q)$ for the q^{th} received symbol of the m'^{th} user are extracted as

$$\begin{aligned}\mathbf{z}_{m'}(q) &= \mathbf{T}_{m'p}^{H}\mathbf{Y}_{m'}(q) \\ &= \mathbf{T}_{m'p}^{H}\mathbf{D}_{\tilde{R}}(\mathbf{h}_{m'}(q))\mathbf{x}_{m'}(q) + \delta_{m'}(q),\end{aligned}$$

where $\delta_{m'}(q) = \mathbf{T}_{m'p}^{H}\mathbf{T}_{m'}^{H}\hat{\mathbf{v}}_{m'}(q)$. Using (4.2),

$$\mathbf{z}_{m'}(q) = \mathbf{T}_{m'p}^{H}\mathbf{D}_{\tilde{R}}(\mathbf{h}_{m'}(q))\left[\mathbf{T}_{m'd}\mathbf{d}_{m'}(q) + \mathbf{T}_{m'p}\mathbf{p}_{m'}(q)\right] + \delta_{m'}(q).$$

As $I_{m'd} \bigcap I_{m'p} = \phi$, $\mathbf{T}_{m'p}^{H}\mathbf{T}_{m'd} = \mathbf{0}$ and $\mathbf{T}_{m'p}^{H}\mathbf{T}_{m'p} = \mathbf{I}_P$, so

$$\mathbf{z}_{m'}(q) = \mathbf{D}_P(\acute{\mathbf{h}}_{m'}(q))\mathbf{p}_{m'}(q) + \delta_{m'}(q).$$

where $\mathbf{D}_P(\acute{\mathbf{h}}_{m'}(q))$ contains the channel frequency response at the pilot sub-carriers. $\mathbf{D}_P(\acute{\mathbf{h}}_{m'}(q))$ is symbol dependent due to hopping, which means that the each pilot sub-carrier $m'R + n_p + n_q$ in consecutive symbols, samples the channel response $\mathbf{h}_{m'}(m'R + n_k + n_q)$ at a different point on the DFT grid. If we collect S consecutive OFDMA symbols, such that $SP > L_m + 1$, we can perform LS or linear minimum mean square error (LMMSE) channel estimation for each user using the received pilot symbols, as in [33, 34].

4.5.3 Residual Phase Compensation

As CFO estimation cannot be perfect, there is always a residual CFO error associated with the CFO estimation which appears as $f_{m'} - \hat{f}_{m'}$ in the compensated signal in (4.14). In SI part of (4.14), $\mathbf{D}_N(f_{m'} - \hat{f}_{m'})$ can be approximated as an identity matrix for each symbol, provided the CFO estimation is accurate enough but the $e^{j\frac{2\pi}{N}(f_{m'} - \hat{f}_{m'})(qN_t + N_{CP})}$ phase component increases as the symbol index increases and severely degrades the bit error rate (BER) performance if not compensated. In order to compensate the effect of this residual CFO error and further improve the CFO estimation performance, we use the non-zero channel estimation pilots to estimate this residual CFO error and feed it back to the compensation block to improve the CFO estimate, as shown in the Figure 4.6. Note that in (4.2), there are other phase components as well, corresponding to the interfering users. However, if we place the non-zero pilots in the middle of the sub-band, like null sub-carriers, we can ignore the effect of these phase components arising due to MUI. We use residual phase compensation in both the preliminary and refined stage.

Assuming that the channel estimation is perfect, we can equalize the channel to get,

$$\begin{aligned}\hat{\mathbf{p}}_m(q) &= \mathbf{D}_P^{-1}(\mathbf{h}_m(q))\hat{\mathbf{z}}_m(q) \\ &= e^{j\frac{2\pi}{N}(f_{m'}-\hat{f}_{m'})(qN_t+N_{CP})}\mathbf{p}_m(q)\end{aligned}$$

where $\hat{\mathbf{z}}_m(q) = \mathbf{T}_{mp}^H\hat{\mathbf{Y}}_{m'}(q)$. As $\mathbf{p}_m(q)$ contains the non-zero pilot sub-carriers known at the BS, we can estimate the residual phase

$$\hat{\theta}(q) = \sum_{q=0}^{S-1}\sum_{i=0}^{P-1}\frac{\hat{p}_m(q,i)}{p_m(q,i)}$$

for each symbol and compensate its effect from $\hat{\mathbf{Y}}_{m'}(q)$. Same residual phase estimation is carried out through $\bar{\mathbf{z}}_m(q)$ to calculate $\bar{\theta}(q)$ and applied to $\bar{\mathbf{Y}}_{m'}(q)$. In addition this residual phase $\bar{\theta}(q)$ can be used to estimate the residual CFO error as

$$f_{m'} - \hat{f}_{m'} = \sum_{q=0}^{S-1}\bar{\theta}(q)\frac{N}{(2\pi(qN_t+N_{cp}))}$$

which is fed back to the CFO estimation block as shown in Figure 4.6 to update the CFO estimate. Simulation results will show that a significant performance improvement can be achieved through this residual phase compensation and CFO error feedback.

4.6 Simulation Results

In this section, we show the performance evaluation of our algorithm and its comparison with the existing approaches in the literature through Monte Carlo simulations. In these simulations we use, $N = 128$, $M_{max} = 8$, $M = 4$, $N_{CP} = \frac{N}{16}$, $K = 1$, $Q = 32$, $P = 1$ and $S = 8$. The modulation scheme is quadrature phase-shift keying (QPSK). Each user's channel is modeled as a Rayleigh fading channel with exponential power delay profile and order $L_m = L = 8$. Two different null sub-carrier positions, i.e., $n_k = 1$ and $n_k = \frac{R}{2}$ have been used. *Case 1 (performance of CFO estimation)*: We show the effect of null sub-carrier position and sub-band hopping in Figure 4.7 which shows the MSE for CFO estimation. Both with and without sub-band hopping cases have been considered. We also compare the performance of our CFO estimation algorithm with [1]. It is clear from Figure 4.7 that sub-band hopping has a significant effect on lowering the MSE and achieving channel independent performance. The effect of null sub-carrier position is also significant. As $n_k = \frac{R}{2}$ is in the middle of the sub-band, it suffers from lower MUI, and thus the MSE for $n_k = \frac{R}{2}$ is less than $n_k = 1$ as well as [1], which shows the superior performance of the proposed algorithm. MSE performance for any sub-carrier position is eventually affected by MUI as the SNR increases but MSE for $n_k = 1$ shows a higher error floor at higher SNR as compared to $n_k = \frac{R}{2}$. Note that our algorithm only uses 1 null sub-carrier per user for CFO estimation while [1] uses the whole guard band (16 null sub-carriers for each user in this case) for CFO estimation, and still our estimation method shows superior results for MSE as compared to [1].

Figure 4.8 shows the effect of increasing the $f_{\max}$ on MSE of CFO estimation. As the maximum normalized CFO increases, the MUI on the null sub-carriers also increases. The effect of increasing

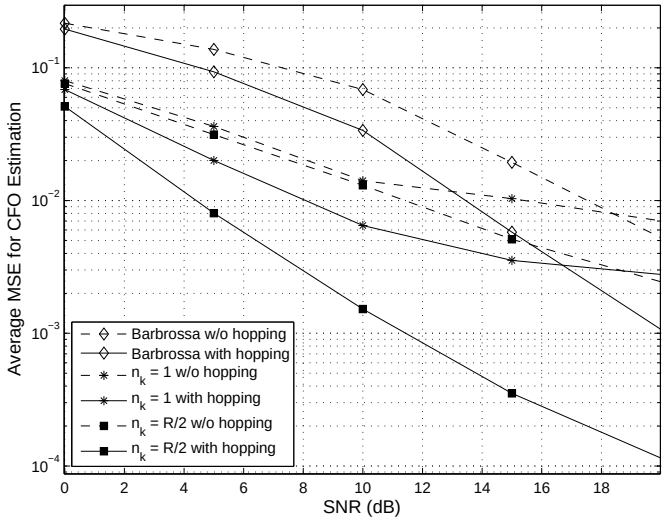


Figure 4.7: Performance comparison of MSE of CFO estimation with Barbrossa *et al.* [1]

$f_{\max}$ is more pronounced on lower SNRs as shown in Figure 4.8 For SNR $>$ 10 dB the effect of increasing $f_{\max}$ is almost negligible.

Figure 4.9 shows the effect of residual CFO error feedback on the MSE of CFO estimation for $n_k = \frac{R}{2}$. The simulation results use only one iteration for the CFO error feedback. As obvious from the Figure the error floor at higher SNRs is significantly improved by updating the CFO estimate based on the CFO error feedback. The error floor can be further reduced by performing more iterations of the CFO error feedback mechanism.

Case 3 (performance of CFO compensation): In Figure 4.10, we compare the performance of DD MUI reconstruction and compensation with the CFO compensation algorithm of [2]. The performance of the system is evaluated at three different normalized CFO values, i.e., $0.5, 1$ and 1.5. As the value of f_{max} increases, the performance of [2] degrades significantly while the degradation in the performance of the DD MUI compensation is negligible. This is because the existing methods cannot support CFO greater than $\frac{1}{2}$ while the proposed DD MUI reconstruction and compensation method supports integer as well as fractional offsets.

4.7 Conclusions

We presented a null sub-carrier based frequency estimation and compensation technique for OFDMA uplink systems. The CFO of each user is estimated before DFT at the receiver by observing the amount of interference at the pre-defined null sub-carrier positions. We investigated the identifiability of the CFO estimation under the effect of MUI. In addition, we discussed the effect of null sub-carrier position and MUI on the acquisition range of the CFO estimation algorithm. We proved that the MUI does not affect the identifiability of CFO estimation under the supported CFO acquisition range. Simulation results demonstrated the superior performance of the proposed algorithm.

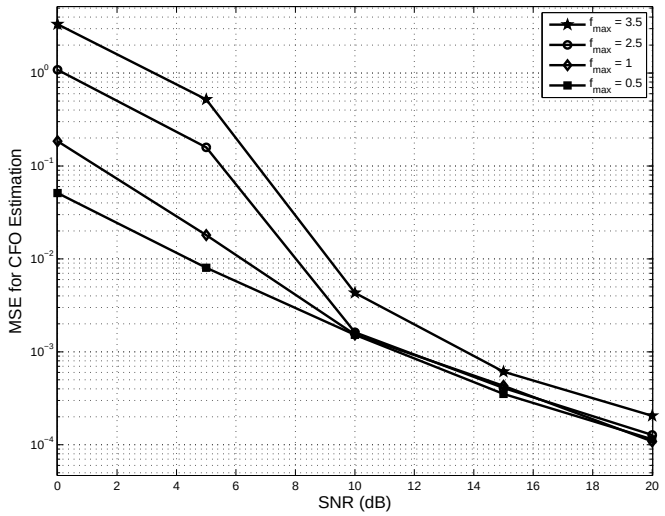


Figure 4.8: Effect of increasing acquisition range

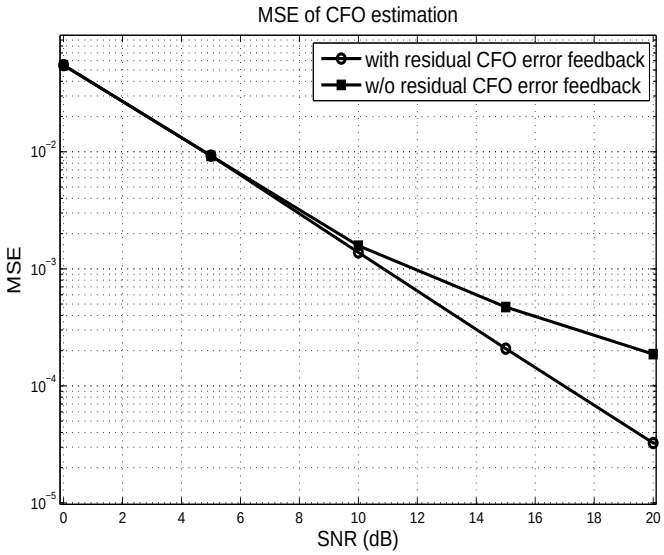


Figure 4.9: MSE of CFO estimation with CFO error feedback

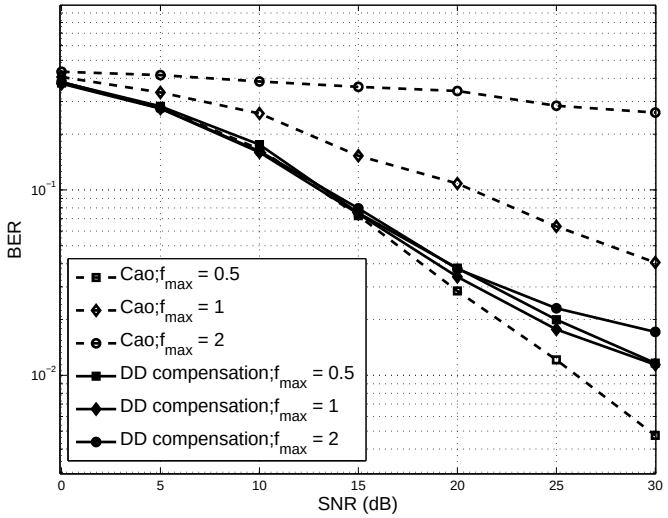


Figure 4.10: BER comparison of DD MUI cancelation scheme with Cao *et al.* [2]

4.8 Appendix: Calculation of worst case curvature through Taylor Series

For the worst case Interference scenario discussed in Section 4.4.3, $f_m - f_o = \pm(2f_{\max} - \frac{1}{2})$ and $f_o = f_{m'}$. We represent the normalized MUI and SI in (4.12), by Taylor series about their own extreme point denoted by a_m, where,

$$a_{m'} = 2\pi\{f_{m'} - f_o + \Delta r\}|_{f_o = f_{m'}} = 2\pi(\Delta r).$$

for the SI term. For the MUI term, when $n_k < \frac{R}{2}$, $f_{\max} = \frac{n_k}{2}$, and

$$a_m = 2\pi\{r - \frac{1}{2} + R\Delta m\}.$$

Similarly

$$a_m = 2\pi\{(\Delta m - 1)R + \frac{3}{2} + r\},$$

for $k \geq \frac{R}{2}$.

Representing normalized MUI in (4.12) through Taylor series at $x = a_m$ gives,

$$\begin{aligned}\mathcal{J}_{m',MUI}(f_o) = \frac{1}{N^2} \sum_{m=0, m\neq m'}^{M_{\max}-1} \sum_{r=0, r\neq n_k}^{R-1} & \frac{1-\cos(a_m)}{1-\cos(\frac{a_m}{N})} + \frac{(x-a_m)}{1-\cos(\frac{a_m}{N})}\left(\sin(a_m) - \frac{1}{N}\frac{\sin(\frac{a_m}{N})(1-\cos(\frac{a_m}{N}))}{(1-\cos(\frac{a_m}{N}))}\right) \\ & + \frac{(x-a_m)^2}{1-\cos(\frac{a_m}{N})}\left(\frac{\cos(a_m)}{2!} - \frac{1}{N^2}\frac{1-\cos(a_m)}{1-\cos(\frac{a_m}{N})}\{\frac{\cos(\frac{a_m}{N})}{2!} - \frac{\sin^2(\frac{a_m}{N})}{1-\cos(\frac{a_m}{N})}\} - \frac{1}{N}\frac{\sin(a_m)\sin(\frac{a_m}{N})}{(1-\cos(\frac{a_m}{N}))}\right) \\ & + \text{higher} \quad \text{order} \quad \text{terms}.\end{aligned}$$

The second derivative of the worst case MUI then becomes,

$$\frac{\partial^2 \mathcal{J}_{m',MUI}(f_o)}{\partial f_o^2} = \frac{8\pi^2}{N^2} \sum_{m=0,m\neq m'}^{M_{\max}-1} \sum_{r=0,r\neq n_k}^{R-1} \{\frac{1}{1-\cos(\frac{a_m}{N})}(\frac{\cos(a_m)}{2} - \frac{1}{N^2}\frac{1-\cos(a_m)}{1-\cos(\frac{a_m}{N})}\{\frac{\cos(\frac{a_m}{N})}{2} - \frac{\sin^2(\frac{a_m}{N})}{1-\cos(\frac{a_m}{N})}\} - \frac{1}{N}\frac{\sin(a_m)\sin(\frac{a_m}{N})}{(1-\cos(\frac{a_m}{N}))})\}$$

Similarly the second derivative of the SI term at its minimum is calculated through Taylor series about its extreme point $a_{m'}$. The curvature of the worst case normalized MUI and SI is given by absolute value of their second derivative at their respective extreme points. Note that all the higher order terms vanish at $x = a_m$, and the second derivative of worst case MUI at $x = a_m$ reduces to

$$\frac{\partial^2 \mathcal{J}_{m',MUI}(f_o)}{\partial f_o^2} = \frac{8\pi^2}{N^2} \sum_{m=0,m\neq m'}^{M_{\max}-1} \sum_{r=0,r\neq n_k}^{R-1} \left(\frac{1}{1-\cos(\frac{a_m}{N})}\{-\frac{1}{2} - \frac{2}{N^2}\frac{1}{1-\cos(\frac{a_m}{N})}(\frac{\cos(\frac{a_m}{N})}{2} - \frac{\sin^2(\frac{a_m}{N})}{1-\cos(\frac{a_m}{N})})\} \right),$$

which reduces to

$$\frac{\partial^2 \mathcal{J}_{m',MUI}(f_o)}{\partial f_o^2} = \frac{8\pi^2}{N^2} \sum_{m=0,m\neq m'}^{M_{\max}-1} \sum_{r=0,r\neq n_k}^{R-1} \{-\frac{1}{2(1-\cos(\frac{a_m}{N}))} + \frac{1}{N^2}\frac{1}{(1-\cos(\frac{a_m}{N}))^2}(2+\cos(\frac{a_m}{N}))\}, \tag{4.15}$$

So,

$$\kappa_{\mathcal{J}_{m',\mathrm{MUI}}(f_o)} = |\frac{\partial^2 \mathcal{J}_{m',MUI}(f_o)}{\partial f_o^2}| < \frac{4\pi^2}{N^2} \sum_{m=0,m\neq m'}^{M_{\max}-1} \sum_{r=0,r\neq n_k}^{R-1} -\frac{1}{1-\cos(\frac{a_m}{N})}.$$

For $n_k < \frac{R}{2}$

$$= \frac{2\pi^2}{N^2} \sum_{m=0,m\neq m'}^{M_{\max}-1} \sum_{r=0,r\neq k}^{R-1} \csc^2\left(\frac{2\pi(r-\frac{1}{2}+(\Delta m)R)}{N}\right)$$
$$< \frac{2\pi^2}{N^2} \sum_{r'=R}^{MR-1} \csc^2(\frac{2\pi(r'-\frac{1}{2})}{N}).$$

As

$$\sum_{n=0}^{N-1} \csc^2\left(\frac{\pi n}{N} + z\right) = N^2 \csc^2(Nz),$$

$$\kappa_{\mathcal{J}_{m',\mathrm{MUI}}(f_o)} = \frac{2\pi^2}{N^2}(N^2 \csc^2(\frac{-\pi}{2}) - \sum_{r'=0}^{R-1} \csc^2(\frac{2\pi r' - \pi}{2N})) < 4.$$

for $R > 2$. Here used the fact that $\csc(x) \approx \frac{1}{x}$ for small x. Similar result is obtained for $n_k \geq \frac{R}{2}$. On the other hand, the curvature of the SI at its minimum becomes,

$$\begin{aligned} \kappa_{\mathcal{J}_{m',\mathrm{SI}}(f_o)} &= \frac{4\pi^2}{N^2} \sum_{r=0, r\neq n_k}^{R-1} \frac{1}{1-\cos(\frac{2\pi(\Delta r)}{N})} \\ &= \frac{2\pi^2}{N^2} \sum_{r=0, r\neq n_k}^{R-1} \csc^2(\frac{2\pi\{r-n_k\}}{2N}) \geq 4, \end{aligned} \tag{4.16}$$

for $R > 2$. Hence it is proved that the curvature of MUI term is always less than the curvature of signal term, at the minimum of the cost function and thus, the identifiability of CFO estimation is always guaranteed.

Chapter 5

Network Modulation: An Algebraic Approach to Enhancing Mesh Network Persistence

Large-scale distributed systems such as sensor networks usually experience dynamic topology changes, data losses, and node failures in various catastrophic or emergent environments. As such, maintaining data persistence in a scalable fashion has become critical and essential for such systems. The existing major efforts such as coding, routing, and traditional modulation all have their own limitations. In this work, we propose a novel *network modulation* (NeMo) approach to significantly improve the data persistence when OFDM is adopted. Built on algebraic number theory, NeMo operates at the level of modulated symbols (so called "modulation over modulation"). Its core notion is to mix data at intermediate network nodes and meanwhile guarantee the symbol recovery at the sink(s) without pre-storing or waiting for other symbols. We also propose subcarrier grouping techniques to enhance the error performance in the presence of fading and additive noises.

5.1 Introduction

Today large-scale distributed systems are routinely deployed for many computing, detection, communication, and monitoring tasks. These systems are comprised of a large number of spatially distributed autonomous devices. It is very common for these systems to incur data losses and node outages. Also because of nodes' random placement, network topology is unknown and the sink location(s) may be unknown. Owing to all of these network uncertainties, how to *safely and soundly* deliver the data to the sink(s) – data persistence – becomes challenging and critical.

There are two major issues which have to be considered and resolved for enhancing data persistence in a large-scale distributed system. One is *how to deliver the existing data to the sink(s) as soon as*

possible. This is an important metric to evaluate the performance of an algorithm targeting data persistence. Routing data to the sink(s) with the minimal transmission overhead (e.g., delay) is a straightforward solution to this issue. However, existing routing protocols such as [35–41] do not work appropriately due to lack of topology information, or they have to pay high communication and storage overhead when nodes are required to initiate data reading and transmission immediately without learning the network topology. The dynamics of network topology and unexpected node failures make things even worse.

The other issue is concerned with *how to "back-up" data in the network so that if one node suddenly fails, its data can still survive in other places of the network*. One natural approach is to adopt coding techniques. Recently, different coding techniques have been proposed (e.g., [42–47]) to increase data persistence. They show great improvement relative to the no coding case. However, there still exist several unsolved problems. For example, some coding techniques require the sink to collect enough packets to decode the next coded packet (see e.g. [48]). This causes extra delay and decoding complexity at the sink and may be impractical for some applications with strict timeliness requirement such as sensor networks for catastrophe monitoring.

In this chapter, we propose a novel approach that is referred to as network modulation (NeMo). NeMo is based on algebraic number theory to enhance data persistence. This approach adopts an algebraic way to "combine" symbols which increases the information in a symbol while still guaranteeing the decodability at the sink. The core notion of NeMo is to mix the data at intermediate network nodes while allowing the sink to decode without pre-stored symbols.

The rest of this chapter is organized as follows. We formulate the problem and describe the network setting in Section 6.3. Section 5.3 introduces the basics of NeMo and the modulation and demodulation steps. We propose subcarrier grouping techniques for OFDM systems to improve the error performance at the sink in Section 5.4 and Section 5.5 evaluates the performance of NeMo. Section 6.8 concludes the chapter.

5.2 Problem Statement

In this section, for simplicity of illustration, we first present a description of a simple network model we will use to describe the design of NeMo. Then, we define data persistence and formulate the problem we attack in this work.

5.2.1 Network Description

Our network model consists of a large sensor network with N sensors/nodes and 1 sink. The network is zero-configuration such that nodes only sense their neighbors with whom they can communicate directly and do not know where the sink is. The network topology is random and can be altered. Typically the majority of the nodes cannot communicate with the sink directly. In addition, we make following assumptions:

- Each node takes only a single reading;
- All data packets have the same importance;

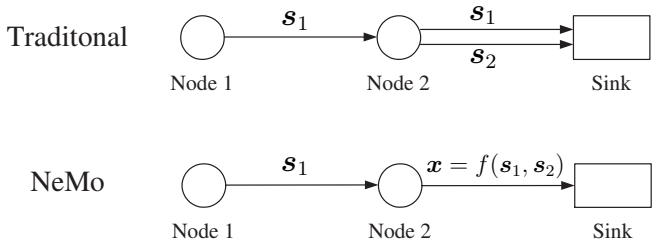


Figure 5.1: A two-node example

- All nodes have the same transmission range;
- Every node employs the same modulation technique and runs the same protocol;
- All nodes have half-duplex capability, i.e., transmitting and receiving at different time slots[1].

The above assumptions construct a simple network model which is most appropriate to show the design principles and facilitate the analysis. Most of the assumptions are also adopted in the literature (see e.g. [40, 47, 49, 50]).

5.2.2 Problem Formulation

Data persistence is defined as the fraction of data generated within the system that eventually reaches the sink [47]. Now let us use a simple example to illustrate what makes NeMo unique to enhance the data persistence.

Example 1: Suppose that there are two nodes (Node 1 and Node 2) with two readings/symbols, s_1 and s_2 for each. The network is two-hop from Node 1 to Node 2 and then to Sink (see Figure 5.1). The goal is to deliver both s_1 and s_2 to the sink. Without combining s_1 and s_2 at Node 2, 3 hops are needed. We can do it in two hops if Node 2 can transmit a combination of s_1 and s_2, $x = f(s_1, s_2)$, in one slot. One question is: *Given two symbols, can we find an efficient approach to combine them as one symbol by guaranteeing identifiability at the sink side?* For example, for BPSK modulated symbols s_1 and s_2, when simple "adding" $x = s_1 + s_2$ is applied, the possible values of x (known as constellation) are shown in the right subfigure of Figure 5.2. The (s_1, s_2) pair to generate x is depicted under the corresponding point of x. From the figure, it is ready to see that the unique recovery of original readings is not guaranteed. For example, if $x = 0$, the sink does not know which pair among $(0, 0)$, $(-1, 1)$ and $(1, -1)$ was sent from Node 1 and Node 2. However, if we "smartly" combine s_1 and s_2 as $x = s_1 + e^{j\pi/4} s_2$, the constellation of x is shown in the left subfigure of Figure 5.2. From the figure, we can see that one unique x is designated to every pair of s_1 and s_2. That means when the sink receives x, it can easily recover the original two symbols s_1 and s_2. This shows that if we combine two symbols "smartly", symbol recovery is guaranteed.

Mathematically, we formulate the problem as follows. Suppose that s_1 is the local symbol at a node and s_2 is a symbol newly received at the node. After linear combination, the symbol transmitted

[1] [47] assumes full-duplex capacity. However, we believe half-duplex is more practical in the context. Our scheme also works for full-duplex scenario.

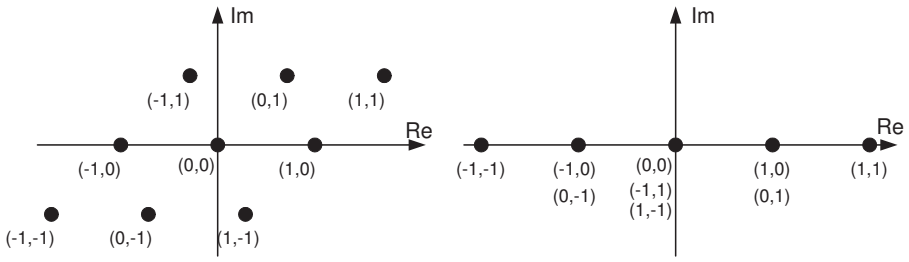


Figure 5.2: Constellation at the sink in a two-node example

from this node to another node or sink is

$$x = \lambda(\theta_1 s_1 + \theta_2 s_2),$$

where λ is the power normalizer, and θ_1 and θ_2 are two coefficients which are specified by modulation schemes. In general, we have

$$x = \lambda \sum_{n=1}^{d} \theta_n s_n = \lambda \boldsymbol{\theta} \boldsymbol{s}^T, \tag{5.1}$$

where $\boldsymbol{\theta} = [\theta_1 \ ... \ \theta_d]$ and $\boldsymbol{s} = [s_1 \ ... \ s_d]$. The remaining question is how to choose $\{\theta_n\}$ so that $\{s_n\}$ can be *uniquely* recovered from x. This may look like an ill-posed problem – given one equation, how can one solve two or more unknowns? The key is that $\{s_n\}$ are not real or complex numbers, but belong to some lattice (e.g. all QAM symbols belong to complex Gaussian integer lattice). By appropriately choosing $\{\theta_n\}$, it can be guaranteed that $\{s_n\}$ will be uniquely identified from x. We give the detailed design in the following sections.

5.3 Design of NeMo

In this section, we briefly introduce algebraic number theory and describe our NeMo design based on the theory.

5.3.1 Terminology and Notation

In the following, we summarize some terminologies and corresponding notations which will be used in the rest of this chapter.

- *Symbol*: We adopt s_k's to denote the originally modulated symbols (before nodes exchange information), e.g., M-ary QAM. We call them an *OM symbols*. Multiple OM symbols can be modulated by NeMo into another symbol x_k called an *NM symbol*.
- *Degree of an NM symbol:* The degree of an NM symbol x is the number of OM symbols employed to generate this symbol and is denoted as d.
- *Maximum degree of an NM symbol:* Due to computational power and memory size constraints, the degree of NM symbols is usually upper bounded. The maximum degree allowed is denoted as $d_{\max}$.

Table 5.1: Design of $\alpha = e^{j2\pi q/P}$

D	2	3	4	5	6	7	8	9	10
P	8	9	16	25	36	49	32	18	50

- *Degree of a modulator:* It is defined as the length of the vector $\boldsymbol{\theta}$ as in (5.1) from which the coefficients θ_n's are drawn and is denoted as D.

5.3.2 Algebraic Number Theory for NeMo

Before we pursue the detailed modulation scheme, we need to introduce some basics of algebraic number theory which will be used to design NeMo.

Euler numbers: Given an integer P, the Euler number $\phi(P)$ of P is the cardinality of the set $\{q : \gcd(q, P) = 1, q \in [1, P)\}$, where gcd stands for the greatest common divisor.

Two properties of Euler numbers are useful [51]:

P1) if P is prime, then $\phi(P^n) = P^{n-1}(P-1)$;

P2) if P, Q are relatively prime, i.e., $\gcd(P, Q) = 1$, then $\phi(P \cdot Q) = \phi(P)\phi(Q)$.

As we mentioned, the key point to design $\boldsymbol{\theta}$ in (5.1) is to make sure that when the OM symbols are linearly combined as an NM symbol, they can still be uniquely demodulated. There are different ways to design $\boldsymbol{\theta}$. Here we are providing a systematic and general way based on algebraic number theory. For a given number of OM symbols d, the design of $\boldsymbol{\theta}$ has the following special structure

$$\boldsymbol{\theta} = [1 \ \alpha \ ... \ \alpha^{d-1}], \tag{5.2}$$

where α is a scalar which will be designed as follows. The general design of α only depends on the modulator's degree D. It does not depend on the original modulation size. For a given modulator degree D, select an integer P which is a multiple of D and $\phi(P) = 2mD$, where m is a positive integer. *The generators α (and thus $\boldsymbol{\theta}$ in (5.2)) can be designed as*

$$\alpha = e^{j2\pi q/P}, \tag{5.3}$$

where q is selected from $[1, P/D)$ such that $gcd(q, P) = 1$, and $j = \sqrt{-1}$. In the following, we provide one example to illustrate the design of α:

Example 2: If $D = 2^k$, $k \in \mathbb{N} \cup \{0\}$, then we can select $P = 2^{k+2} = 4D$, and the Euler number $\phi(P) = 2D$ (c.f. property P2)). We can choose $q = 1$ such that $\gcd(q, P) = 1$. Hence, $\alpha = e^{j2\pi/P}$.

Note that the choice of α is not unique. Different α choices for the same size D may provide different performance in physical layer (see e.g. [51]), but all of them achieve the same symbol identifiability. In Table 5.1, we list the design of α with some commonly used values of D. Although the choice of q is non-unique, *in the following, we adopt the universal choice for all D, i.e., $q = 1$.*

5.3.3 The Basics of NeMo

Now we are ready to go into the design of NeMo. Note that, for simplicity we assume that time is divided into rounds as in [47]. In each round, a pair of nodes completes a packet exchange if no collision happens. The basic procedure works as follows.

- (*Initialization*) Every node has one packet ready if any.
- (*Exchange*) In each round, each node transmits its packet with probability p.
  a) If a node decides to transmit the packet, it will randomly select a neighbor to forward the packet. The selected neighbor will receive the packet if it does not transmit in the meanwhile (recall that we assume half-duplex channel.). Otherwise the packet is dropped and the rest of the round becomes idle. Collision may also happen if a node is chosen for exchange by more than one neighboring nodes at the beginning of a round.
  b) Those nodes which successfully received packets will forward their stored packets back to the corresponding nodes to complete an exchange round.

The above procedure works iteratively and after some rounds the full data persistence will be achieved at the sink. Next we will describe how to modulate and process incoming packets in detail.

5.3.4 Network Modulation and Demodulation for OFDM Systems

We adopt OFDM systems with K subcarriers for the network model described in Section 5.2.1. When a node decides to exchange packets, it modulates K NM symbols from m OM symbols stored in its memory and assigns each NM symbol to one subcarrier. If the number of OM symbols stored in the node's memory m is less than K, OM symbols are not modulated and the node assigns each OM symbol to one of the available subcarriers. Otherwise, the node modulates m OM symbols into K NM symbols as shown in the next example.

Example 3: If $K = 16$ and the number of stored OM symbols is $m = 35$, then 3 subcarriers are assigned 3 OM symbols each and the rest 13 subcarriers are assigned 2 OM symbols each so that the number of OM symbols assigned is as evenly distributed as possible across subcarriers. The modulator $\boldsymbol{\theta}$ with a modulator degree $D \geq 3$ is adopted to modulate OM symbols as

$$x = \lambda \boldsymbol{\theta} \boldsymbol{s}^T \tag{5.4}$$

for each subcarrier, where $\boldsymbol{s} = [s_1 \ ... \ s_d]$ is a vector of assigned OM symbols. Thus, the node now has $K = 16$ NM symbols modulated from $m = 35$ OM symbols.

For the demodulation, note that the demodulation complexity is determined by the degree of each NM symbol d. Therefore, a non-sink node may not be able to afford demodulating NM symbols with unbounded degrees and have to set a constraint on the maximal degree (denoted by $d_{\max}$) of an NM symbol by setting the modulator degree of $\boldsymbol{\theta}$ in (5.4) to $D = d_{\max}$. Since K NM symbols are exchanged in a round, up to $Kd_{\max}$ OM symbols are exchanged in a round and it is denoted by T. As $d_{\max}$ increases, more OM symbols can be exchanged in a round and we expect that data persistence will be enhanced. However, the error performance at the sink will become worse since more OM symbols are packed in one NM symbol, which reduces the Euclidean distance of constellation points. We will

verify this in the simulations later.

After receiving K NM symbols after exchanging, a node demodulates each NM symbol (with degrees, say, d) into d OM symbols by exhaustively searching over all possible $d \times 1$ OM symbol vectors. Due to the design of $\boldsymbol{\theta}$ in Section 5.3.2, a $d \times 1$ OM symbol vector can be uniquely determined for each NM symbol. After demodulating all K NM symbols, the node just stores demodulated OM symbols into its memory if it is a sink node. For non-sink nodes, before explaining how to store demodulated OM symbols, let us define an important concept – the effective degree of demodulated OM symbols first. The set of OM symbols already stored in the node's memory is denoted as $\mathcal{X}$. The *effective degree* of demodulated OM symbols, denoted by d_e, is defined as the number of OM symbols that are not present in $\mathcal{X}$. Suppose the effective degree of demodulated OM symbols is d_e and the node has m $(1 \leq m \leq T)$ OM symbols stored. Then, the node stores demodulated OM symbols as follows:

- If $d_e = T$, the node randomly replaces one OM symbol among the d_e OM symbols received with its local reading. These T OM symbols will be saved in the memory and other symbols in the memory will be discarded.
- If $d_e < T$ and $d_e + m > T$, the node will randomly pick $T - d_e - 1$ symbols from the $m - 1$ non-local OM symbols, and save them with the local reading and d_e newly received OM symbols. The other OM symbols in the memory are discarded.
- Otherwise $(d_e + m \leq T)$, the node just saves $d_e + m$ OM symbols in the memory.

In any case, the node only stores up to T OM symbols in its memory.

In summary, the general rules for NeMo are: (i) giving the newly received OM symbols and local reading higher priority to be stored and transmitted in the next round so that the new data has more chances to be circulated as soon as possible; and (ii) discarding the old OM symbols in the memory in order to limit space complexity.

5.4 Improving the Error Performance of NeMo with OFDM

In the presence of fading and additive noise, the error performance at the sink becomes worse as $d_{\max}$ increases since more OM symbols are packed in an NM symbol. In this section, we propose to extend a modulator $\boldsymbol{\theta}$ to a modulator matrix $\boldsymbol{\Theta}$ to generate a group of NM symbols instead of just one NM symbol to improve the error performance. First, we will see how we can extend $\boldsymbol{\theta}$ to a modulator matrix $\boldsymbol{\Theta}$.

5.4.1 A Modulator Matrix $\boldsymbol{\Theta}$

As discussed in Section 5.3.2, for a given modulator degree D, the design of generators α (and thus $\boldsymbol{\theta}$ in (5.2)) is non-unique depending on the choice of q as shown in (5.3). Suppose $\alpha_1, \cdots, \alpha_D$ were

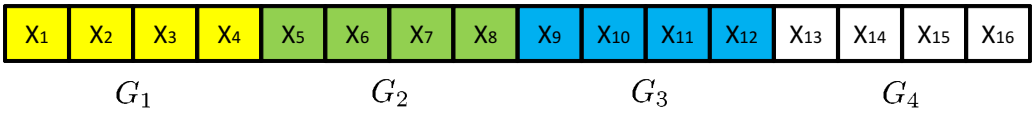


Figure 5.3: Grouping OFDM subcarriers

generated with different values of q. Then, we can form a modulator matrix $\mathbf{\Theta}$ as

$$\mathbf{\Theta} = \begin{bmatrix} 1 & \alpha_1 & \cdots & \alpha_1^{D-1} \\ 1 & \alpha_2 & \cdots & \alpha_2^{D-1} \\ \vdots & \vdots & & \vdots \\ 1 & \alpha_D & \cdots & \alpha_D^{D-1} \end{bmatrix}, \tag{5.5}$$

where $\boldsymbol{\theta}$ can be obtained from by choosing one of the row vectors. From this modulator matrix $\mathbf{\Theta}$, we modulate D OM symbols $\boldsymbol{s} = [s_1 \ ... \ s_D]$ to get D different NM symbols $\boldsymbol{x} = [x_1 \ ... \ x_D]$ as follows,

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_D \end{bmatrix} = \lambda \mathbf{\Theta} \boldsymbol{s}^T = \lambda \begin{bmatrix} 1 & \alpha_1 & \cdots & \alpha_1^{D-1} \\ 1 & \alpha_2 & \cdots & \alpha_2^{D-1} \\ \vdots & \vdots & & \vdots \\ 1 & \alpha_D & \cdots & \alpha_D^{D-1} \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \\ \vdots \\ s_D \end{bmatrix}.$$

Note that, every x_i has the same identifiability, i.e., we can uniquely decode $\boldsymbol{s} = [s_1 \ ... \ s_D]$ from each of $\boldsymbol{x} = [x_1 \ ... \ x_D]$.

5.4.2 Subcarrier Grouping Techniques for OFDM Systems

Now we introduce subcarrier grouping techniques for OFDM systems to improve the error performance at the sink. Suppose we have K subcarriers available. We divide K subcarriers into G groups such that each group has M subcarriers. For example, let $K = 16$, $G = 4$ and $M = 4$ (see Figure 5.3). The same color represents the same group. Note that, this is a subband assignment scheme but any carrier assignment scheme can be applied. X_i is an NM symbol carried by i^{th} subcarrier and we assume i.i.d. subcarrier channel gains. Then, how should we design $\{X_i\}_{i=1}^K$ to improve the error performance at the sink?

If a node decides to transmit, OM symbols stored in a memory are equally divided and assigned to each group similarly as in Example 3. Note that the maximum number of OM symbols exchanged in one round is denoted by T and thus each group is assigned up to T/G OM symbols. Next, the node constructs a $D \times D$ modulator matrix $\mathbf{\Theta}$ (5.5), where D is set to T/G. Then, using this modulator matrix, a set of M NM symbols are generated from the OM symbols assigned in the same group. We give an example to illustrate this.

Example 4: Suppose $K = 16$, $G = 4$, $M = 4$ and $T = 16$. If a node has 10 OM symbols $\{s_n\}_{n=1}^{10}$ to transmit, it assigns OM symbols to each group as equally as possible like $\{s_1, s_2, s_3\} \in G_1$, $\{s_4, s_5, s_6\} \in G_2$, $\{s_7, s_8\} \in G_3$, and $\{s_9, s_{10}\} \in G_4$. Then, we construct a $D \times D$ modulator matrix $\mathbf{\Theta}$

as

$$\mathbf{\Theta} = \begin{bmatrix} 1 & \alpha_1 & \alpha_1^2 & \alpha_1^3 \\ 1 & \alpha_2 & \alpha_2^2 & \alpha_2^3 \\ 1 & \alpha_3 & \alpha_3^2 & \alpha_3^3 \\ 1 & \alpha_4 & \alpha_4^2 & \alpha_4^3 \end{bmatrix},$$

where $D = T/G = 4$. Using this modulator, we obtain a set of NM symbols from the assigned OM symbols for each group. The submatrix of $\mathbf{\Theta}$ is used to modulate NM symbols. For example, if a group contains less than $D = 4$ OM symbols, NM symbols are generated as

$$\begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{bmatrix} = \lambda \begin{bmatrix} 1 & \alpha_1 & \alpha_1^2 \\ 1 & \alpha_2 & \alpha_2^2 \\ 1 & \alpha_3 & \alpha_3^2 \\ 1 & \alpha_4 & \alpha_4^2 \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \\ s_3 \end{bmatrix}, \quad \cdots \quad , \quad \begin{bmatrix} X_{13} \\ X_{14} \\ X_{15} \\ X_{16} \end{bmatrix} = \lambda \begin{bmatrix} 1 & \alpha_1 \\ 1 & \alpha_2 \\ 1 & \alpha_3 \\ 1 & \alpha_4 \end{bmatrix} \begin{bmatrix} s_9 \\ s_{10} \end{bmatrix}.$$

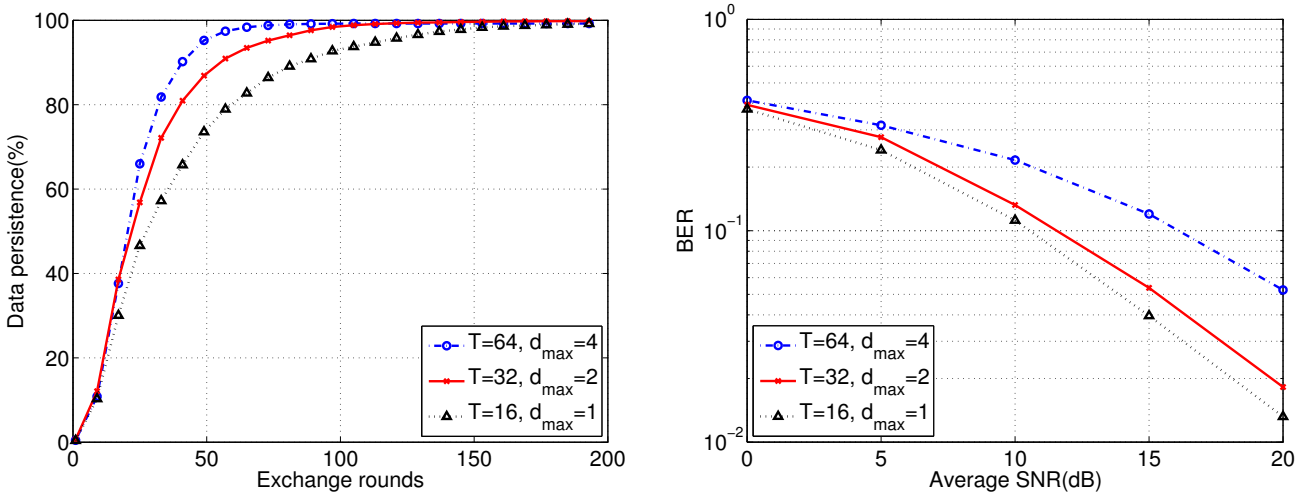
The demodulation process is similar to that in Section 5.3.4. The difference is that the demodulation here is performed on a per-group basis instead of on a per-subcarrier basis. The error performance is enhanced due to the diversity collection at the receiver which is enabled by a modulator matrix (see [51–53] for other applications). Note that Section 5.3.4 is a special case with $M = 1$.

5.5 Performance Evaluation

In this section, we adopt computer simulations to evaluate the performance of NeMo. The numerical evaluation setup for our NeMo technique is described as follows. The network is generated by randomly distributing $N = 100$ nodes in a 1×1 square area. One sink node is also randomly placed in the network to collect the information but does not generate its own reading. The radius of the neighborhood for each of these 101 nodes is $R = 0.3$. BPSK modulation is employed for OM symbols generated at each node. The probability that each node transmits its packet at the beginning of each exchange round is fixed as $p = 0.5$. Also, $K = 16$ subcarriers are available to each node.

Based on this network setup, we show the data persistence obtained by simulating our NeMo with different $d_{\max}$ in Figure 5.4a. For larger $d_{\max}$, the persistence curve increases faster since the maximum number of OM symbols exchanged in a round ($T = Kd_{\max}$) also increases. Note that $d_{\max} = 1$ corresponds to the no coding scheme. However, as discussed in Section 5.3.4, the error performance at the sink becomes worse as T is increased since more OM symbols are packed in one NM symbol, which reduces the Euclidean distances between constellation points. This is verified in Figure 5.4b.

The subcarrier grouping techniques for NeMo are proposed in Section 5.4 to improve the error performance at the sink. Here, we show the BER performance of NeMo with subcarrier grouping techniques with different group size M for $T = 16$ and $T = 32$ in Figure 5.5. The larger the group size M is, the more diversity is enabled by modulating/demodulating on a per-group basis. Thus, the BER curve decays faster for large M.



(a) Data persistence without subcarrier grouping for $K = 16$ (b) BER at the sink without subcarrier grouping $K = 16$

Figure 5.4: NeMo without subcarrier grouping

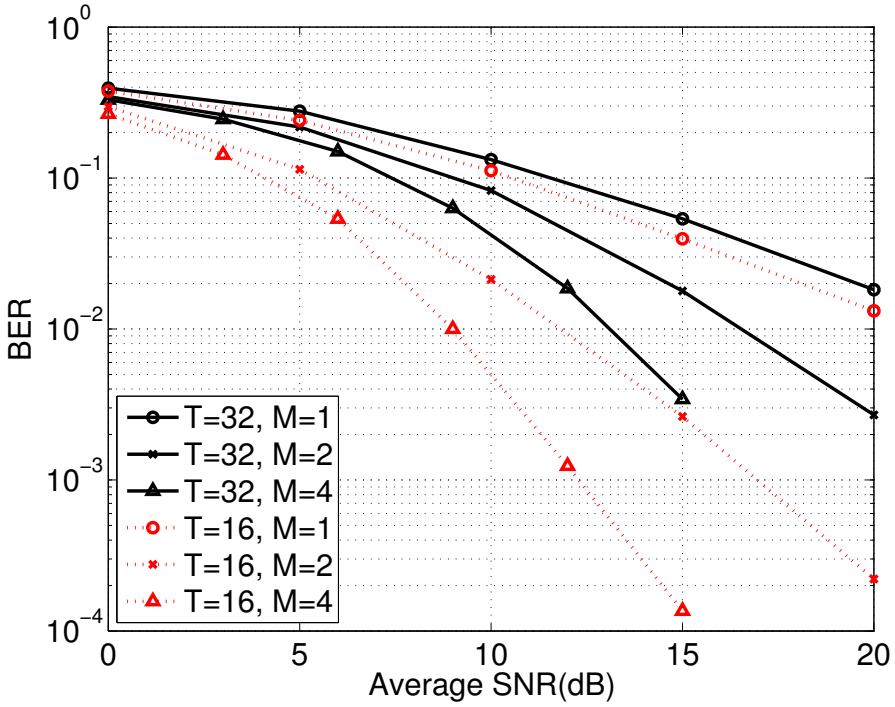
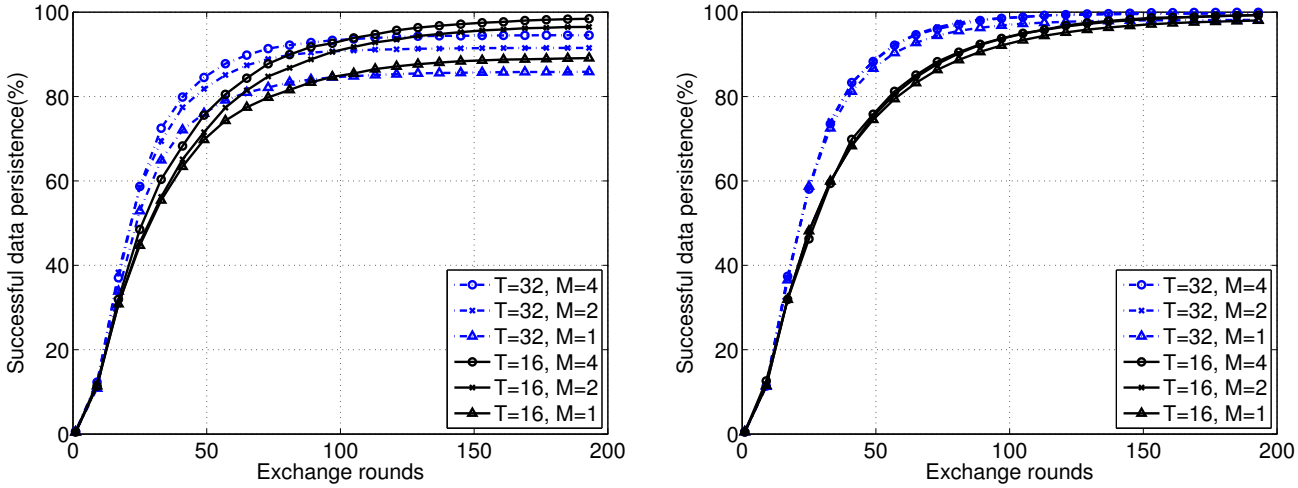


Figure 5.5: BER performance with subcarrier grouping with different (T,M)

Since the error rate is not negligible when the SNR is low, we define another performance metric, called "successful data persistence", to incorporate the effect of the error rate. It is defined as "the fraction of 'correct' data that eventually reaches the sink". We only consider correctly decoded OM symbols when measuring data persistence. Now, we compare the successful data persistence for low (10dB) and high (20dB) SNR settings in Figure 5.6a and Figure 5.6b, respectively. As we can see in the figures, even though T is fixed, the successful data persistence depends on the group size M, especially when the SNR is low, since the BER depends on M as we have seen in Figure 5.5. Also, we



(a) SNR=10dB (b) SNR=20dB

Figure 5.6: Data persistence with subcarrier grouping for $K = 16$

observe that $T = 16$ curves outperform $T = 32$ curves in Figure 5.6a since as we have seen in Figure 5.4b, large T leads to the increased BER. When the SNR is high, the effect of changing M is becomes small.

5.6 Conclusion

In this chapter, we have proposed a new approach – network modulation (NeMo) to significantly enhance data persistence for large-scale distributed systems with OFDM. Based on algebraic number theory, NeMo mixes data at intermediate network nodes and meanwhile guarantees the symbol recovery at the sink(s). Also, we have proposed subcarrier grouping techniques to enhance the error performance at the sink. Simulation results verify our claims.

Chapter 6

Tracking Low-Precision Clocks with Time-Varying Drifts Using Kalman Filtering

Clock synchronization is essential for a large number of applications ranging from wired network measurements to data fusion in sensor networks. Earlier techniques are either limited to undesirable accuracy or rely on specific hardware characteristics that may not be available for certain systems. In this work, we examine the clock synchronization problem in networks where nodes lack the high accuracy oscillators or programmable network interfaces some previous protocols depend on. This chapter derives a general model for clock offset and skew and demonstrates its applicability to real clock oscillators. We design an efficient algorithm based on this model to achieve high synchronization accuracy. This algorithm applies the Kalman filter to track the clock offset and skew. We demonstrate the performance advantages of our schemes through extensive simulations and real clock oscillator measurements. Furthermore, we propose a skew estimation technique without exchanging timestamps and corroborate it through simulations.

6.1 Introduction

The availability of an accurately synchronized clock enables and enhances a wide range of applications in distributed environments that facilitate pervasive computing and communications. For example, Internet measurements, which rely on either passively monitoring network events (e.g., packet loss) or actively probing network conditions (e.g., end-to-end delay and loss rate), implicitly require a common notion of time among all participating measurement points. Another example lies in Wireless Sensor Networks (WSNs). Sensor network applications need a common notion of time for precise data integration and sensor reading fusion. Clock synchronization is also essential in network and

communications protocols such as TDMA medium access scheduling, power mode energy saving, and scheduling for directional antenna reception.

Many clock synchronization techniques for the Internet have been proposed over the past few decades, among which the most popular and widely used is Network Time Protocol (NTP). The development and evolution of NTP are described in Mills' classic chapters [54, 55]. Several techniques [56–59] have been proposed to improve its synchronization accuracy when NTP is not able to satisfy the requirements of demanding applications. Additionally, in applications that do not require real-time synchronization, several other techniques are proposed to estimate and remove clock offset and skew offline in captured data sets such as a packet delay trace [60, 61].

In the Internet, each node is either a router or a host which is wired to a constant power source and has one or more stable and powerful CPU's. In contrast, some other networks have only very limited resources such as scarce energy, unstable processors, and unreliable and low communication bandwidth. WSN is a representative example of this type of networks. In a WSN, the vast majority of sensors are battery-powered so that a desirable clock synchronization scheme must preserve energy to prolong the battery life. Pottie et al. [62] shows that transmitting 1 bit over 100 meters requires 3 joules, which can be used for executing 3 million instructions. Therefore a successful clock synchronization scheme must minimize the amount of message exchange and at the same time maintain high synchronization accuracy. Scarcity of power on sensor nodes however is not the only resource constraint. Due to its small size and low cost, the clock readings in a sensor are derived from oscillators with only limited stability (due to phase noise, thermal noise, aging, etc). Consequently, clocks on sensors are easily affected by temperature variations, vibration and interference and can significantly deviate from the reference sources [**?**, 63]. The situation could become even worse under catastrophic conditions such as earthquake, battlefield, and forrest fire. All these affect the clock drifting rates and make the clock drift nonlinear and time-varying.

Many clock synchronization techniques, including Network Time Protocol (NTP) [54, 55] require to exchange timestamps among nodes for clock synchronization [64–66]. In most practical systems, timestamps are put in the header/payload of a message, which means that only finite bits can be allocated for synchronization purpose. While the small number of bits allocated to timestamps raises communication bandwidth efficiency, the resulting timestamp could deviate significantly from the original real value and degrade the synchronization performance. At the same time, transmitting a long sequence also reduces the energy efficiency. Therefore, a clock synchronization scheme in WSNs needs to minimize the amount of message exchange and while maintaining high synchronization accuracy.

In this work, we design clock estimation and synchronization techniques that work on low precision oscillators with time-varying drift rates.

First, we decompose the clock uncertainty into multiple independent components and use these components to construct general models for a real clock. The proposed models are also general enough to subsume the existing models. Additionally, we introduce techniques to determine the number of parameters of the models.

Second, a Kalman filter is designed to track the clock uncertainty based on the aforementioned models. In fact, most of the prior protocols fail when the clock has some time-varying drift rate

performance. We model the random drift of the clock using Kalman filter which tracks the variation of the clock drift and thus enhances the synchronization performance.

Third, we measure two low-cost oscillators and then apply Occam's razor principle on the measured data to derive the clock model. It shows that Kalman filtering design tracks the real clock drifting well.

Fourth, we further evaluate the clock model and the tracking method by considering missing or corrupt observations which can be occurred in unreliable links. Interestingly, we find that thanks to the success track of our model, the Kalman filter based clock tracking method is not only robust to missing data but also matched well with least absolute shrinkage and selection operator (LASSO) to detect the corrupt data.

Fifth, we propose a clock skew estimation technique without transmitting timestamps, which are still able to achieve desirable synchronization with minimal communication bandwidth.

Note that our clock synchronization scheme establishes local clock models and a Kalman filter based tracking algorithm, which does not require any particular message exchange mode. Therefore, our scheme can be easily adapted into both receiver-to-receiver [64, 67–69] and sender-to-receiver modes [65, 66, 70–73].

The rest of the chapter is organized as follows. We first summarize the related work in Section 6.2. Section 6.3 describes the autoregressive models we develop for low-precision clocks. In Section 6.4 we present and analyze our algorithms to track clock skew and offset. In Section 6.5, we demonstrate our tracking method with the real measurements. We examine the clock tracking performance in systems with missing or corrupt observations in Section 6.6. In Section **??**, a clock skew estimation method is suggested and the performance is compared with existing methods. We conclude the chapter in Section 6.8.

6.2 Related Work

Clock synchronization mechanisms ensure that physically dispersed processors have a common knowledge of time. This topic is well studied in wired networks such as the global Internet. The most common and widely used mechanism is NTP [54, 55], which uses NTP packets containing timestamp information exchanged between NTP server and the host across a network to perform time synchronization. NTP is designed to provide clock offset accuracy bounded by the round-trip time (RTT) between the server and the client.

However, NTP provides insufficient accuracy and robustness for many demanding applications. A few techniques have been proposed to improve measurement accuracy or clock stability. In [56] the synchronization is offloaded into a programmable network interface card. This card would autonomously perform synchronization by sending periodic messages and perform timestamping when packets arrive. In [57, 58], the methods of using a clock based on the more accurate CPU oscillator were proposed. This method relies on the high reliability of the processor oscillator and the availability of a TimeStamp Counter (TSC) register. The Precision Time Protocol [59] was drafted into the IEEE 1588 standard for synchronization of network measurement and control systems. It uses a specially

designed network infrastructure to achieve high synchronization accuracy. Unfortunately, all these techniques may not be effective for low-cost devices.

Since some passive network monitoring tasks do not require real-time synchronization, a few algorithms have emerged for synchronizing data captures. In [60] a linear-programming based algorithm for estimating and removing the skew and offset of a data set was proposed. Convex hulls were used in [61] to estimate the clock skew and offset within a dataset. This was shown to perform better than the linear regression methods, but suffers increased computational complexity. All these algorithms are offline, which means they deal with the saved measurement data such as network packet delay traces instead of online synchronizing the clock.

Most of existing methods synchronize a sender with a receiver by transmitting the current clock values as timestamps [65, 66, 70–73]. In this regard, these methods are vulnerable to variance in message delay between the sender and the receiver due to network delays and the involved workloads. Some other methods [64, 67–69] perform receiver-to-receiver synchronization. These methods exploit the property of the physical broadcast medium where any receivers one-hop away receive the same message at approximately the same time. Such an approach reduces message delay variance due to the reduced time-critical path which is the path of a message that contributes to non-deterministic errors. Our proposed scheme is independent of the above synchronization modes. It can be easily adapted into both modes as far as we obtain reasonably good parameter estimates.

Kalman filtering has been used in the context of clock synchronization [74–76] for packet-switched networks. In [74] Kalman filter was used to model the packet jitter after shaping its characteristics by low-pass prefiltering. [75] presents a Kalman filtering algorithm for end-to-end time synchronization. This algorithm assumes the constant clock skew in a long term, which is not valid in most resource-constrained networks, and relies on NTP to exchange timestamp information. [76] also assumes the constant clock skew and relies on TSC register, found in Pentium class PC's to count CPU cycles.

6.3 Clock Modeling

Network time synchronization at its simplest is not a difficult problem to understand. It is simply the problem of setting two or more clocks with the same notion of time, and performing updates to ensure this continues to occur. This problem becomes complicated however, when the characteristics of the network and clocks themselves are considered. Information sent over a network is subject to random, variable delays which add significant measurement noise to time measurements. Oscillators in clocks suffer from skew, drift and jitter. All of these cause the clocks to progress somewhat erratically. In this section, we carefully study these characteristics. Let us first demonstrate some important terms used in this chapter. Then we show how to model the uncertainty of a clock.

6.3.1 Terminology

As we have mentioned, this work takes the unique viewpoint from physical (hardware) perspective and addresses what causes clock drifting and how to model it.

- **Oscillator**: An oscillator is an electronic circuit that produces a periodic electronic signal, often a sinusoidal waveform. Oscillators are important in many different types of electronic equipment. For example, a quartz watch uses a quartz oscillator to keep track of time. An AM radio transmitter uses an oscillator to create the carrier wave for the station, and an AM radio receiver uses a special form of oscillator called a resonator to tune to a station. There are oscillators in computers, sensors, metal detectors and even stun guns.
- **Phase Noise**: Phase noise is a common type of noise existing in oscillators. An ideal oscillator would generate a pure sinusoid waveform. However, because of time domain instabilities (e.g., "jitter"), the frequency generated cannot be restricted to a single sine. That means the phase noise components spread the power of the signal to adjacent frequencies and the frequency domain representation of the signal shows some rapid, short-term, random fluctuations in the phase.
- **Clock**: A clock is a device that measures time. It generally consists of a periodic component (e.g., an oscillator) and a counting component (e.g., a hardware register). Their combination determines the resolution (i.e., the smallest measurable time unit), and accuracy.
- **Clock Drift**: Clock drift refers to some related phenomena where a clock does not run at the correct speed compared to the actual time. The phase noise in oscillators is an important component of clock drift. Because phase noise is random, the clock drift is also random. Clocks often drift differently depending on their oscillator quality, the exact power they get from the battery, temperature, pressure, humidity, age and so on. Thus the same clock could have different clock drift rates at different occasions. Usually the instantaneous clock drift rate is called *clock skew* and the time difference with the actual time is called *clock offset*. Their formal definitions follow.

6.3.2 Clock Offset Modeling

A clock is merely the combination of an oscillator and a counter. The characteristics of the oscillator and the counter define the clock's behavior. The starting values of the counters control the initial relative offset between clocks. The frequency of the oscillator controls the rate that the clock advances. Since it is impossible to create oscillators that oscillate at exactly the same rate, every clock advances at a different rate in real world. Considering all these factors, we define and model the clock offset next.

Continuous-Time General Clock Model

The time reported by a clock at some ideal time t is written as $C(t)$. We will write $C_A(t)$ as the time given by clock A at time t. The difference between the time of an ideal clock and a given clock is said to be the offset, $\theta(t)$, which is defined as:

$$\theta(t) = C(t) - t.$$

The relative offset from node B to node A, $\theta_A^B(t)$, is defined as:

$$\theta_A^B(t) = C_A(t) - C_B(t) = \theta_A(t) - \theta_B(t).$$

The oscillator in a clock produces periodic pulses. The difference between the rate these pulses are produced and the rate an ideal clock counts the desired interval is called the skew denoted by α:

$$\alpha(t) = \frac{d\theta(t)}{dt} \approx \frac{\theta(t+\tau) - \theta(t)}{\tau}. \tag{6.1}$$

The skew of a clock is the slope of the change in offset compared to the ideal clock. The slope of the relative offset $\theta_A^B(t)$ is relative skew $\alpha_A^B(t)$. This is defined as:

$$\alpha_A^B(t) = \alpha_A(t) - \alpha_B(t).$$

If the oscillator were perfectly stable, the slope of $\theta(t)$ would reflect a constant skew α. However, this is not the case, especially in low-cost devices. Oscillators do not produce perfectly periodic pulses. The oscillator's nonlinearity and the phase noise alter the pulse period, making the clock rate time-varying [77]. Additionally, physical effects such as temperature and age can change the oscillator frequency. For the remainder of the chapter, we assume the reference node has a perfect clock (i.e., zero offset and skew) so that all the offset and skew notations lack the subscripts without loss of generality. It is straightforward to adapt all of derivations into the relative sense when the reference node deviates from the actual time.

Having a complete understanding of clock drift, we decompose its variations into three independent components: the instantaneous clock skew $\alpha(t)$, the initial clock offset θ_0, and the random measurement and other types of additive noise $w(t)$. The instantaneous clock offset $\theta(t)$ at time t is given as:

$$\theta(t) = \int_0^t \alpha(\tau)\ d\tau + \theta_0 + w(t). \tag{6.2}$$

This model is quite general and subsumes all those existing simpler clock models. For example, if the clock skew $\alpha(t)$ does not change along with time t, the model in (6.2) reduces to the simple skew model in [57].

Discrete-Time Clock Model

After sampling, the continuous-time model becomes discrete-time model. In most cases a discrete clock model is desirable since the synchronization is typically achieved by timestamped message exchange. The timestamps are nothing but discrete samples of the continuous time. Based on (6.2), the discrete-time clock model is obtained as:

$$\theta[n] = \sum_{k=1}^{n} \alpha[k]\tau[k] + \theta_0 + w[n], \tag{6.3}$$

where k is the sample index, "$[\cdot]$" is adopted for discrete indexing, $\tau[k]$ is the sampling period at the k^{th} sample.

Here note that our discrete-time model is also quite general. It covers not only uniform sampling, but also non-uniform sampling (by choosing different $\tau[k]$). Since $w[n]$ is mainly caused by the obser-

vation and measurement noise, it is reasonable to assume $w[n]$'s are independently distributed with variance σ_w^2. The variance of $w[n]$ depends on the time-critical path [64]. In general, the time-critical path in a sender-to-receiver synchronization consists of four factors [78]: i) the time for message construction and sender's system overhead, ii) the time to access the transmit channel, iii) propagation delay, and iv) the time spent by the receiver to process the message. In contrast, a receiver-to-receiver synchronization is only impacted by iii) and iv) and hence has smaller variance. In either case, the variance value can be estimated using samples.

We can rewrite this model using a recursive form as:

$$\theta[n] = \theta[n-1] + \alpha[n]\tau[n] + v[n], \tag{6.4}$$

where $v[n] = w[n] - w[n-1]$. Clearly, $v[n]$ is a random variable with mean 0 and variance $\sigma_v^2 = 2\sigma_w^2$. This is not surprising since (6.4) is the differential form of the observation equation for the clock, and differential forms are well-known to double the noise variance. If the observation noise $w[n]$ has non-zero mean, because of the differential format in (6.4), it is not difficult to verify that $v[n]$ still has zero mean.

6.3.3 Clock Skew Modeling

When using the recursive model in (6.4), to synchronize the clock, we need to estimate the clock skew $\alpha[n]$ which is also time-varying in this chapter. Before we establish the clock skew model, we look at two extreme cases of clock skew.

Case i (constant skew): Suppose the clock skew $\alpha[n]$ is constant as in [57, 60]. From (6.4), since $\theta[k]$'s are known for $k = 1, \ldots, n$, if the sampling period $\tau[k]$'s are also known, the optimal clock skew estimator (in terms of mean-square error (MSE)) is

$$\hat{\alpha}[n] = \frac{\sum_{k=2}^{n}(\theta[k] - \theta[k-1])\tau[k]}{\sum_{k=2}^{n}\tau^2[k]}. \tag{6.5}$$

Case ii (independent skew): If the clock skew $\alpha[n]$ changes completely from one sample to another, the optimal estimator becomes

$$\hat{\alpha}[n] = \frac{\theta[n] - \theta[n-1]}{\tau[n]}. \tag{6.6}$$

These two cases are simple, but neither of them is practical. Most of the existing schemes are based on these two simple cases without considering any statistical and time-series model of the clock skew. Because of the phase noise of oscillator, the clock skew has certain randomness, but is not completely independent for each sample. Fig. 6.1 is an example of the real clock skew behaviors in resource-constrained networks. It shows the skews of the two clocks used in a low-powered micro-controller platform, which are 32.768kHz and 16MHz. We examined the clock skews using the platform in a temperature controlled room over 1.5 months. The variation of skews is observable over small timescales and it is clear that the constant skew model fails over hour timescale even in air controlled

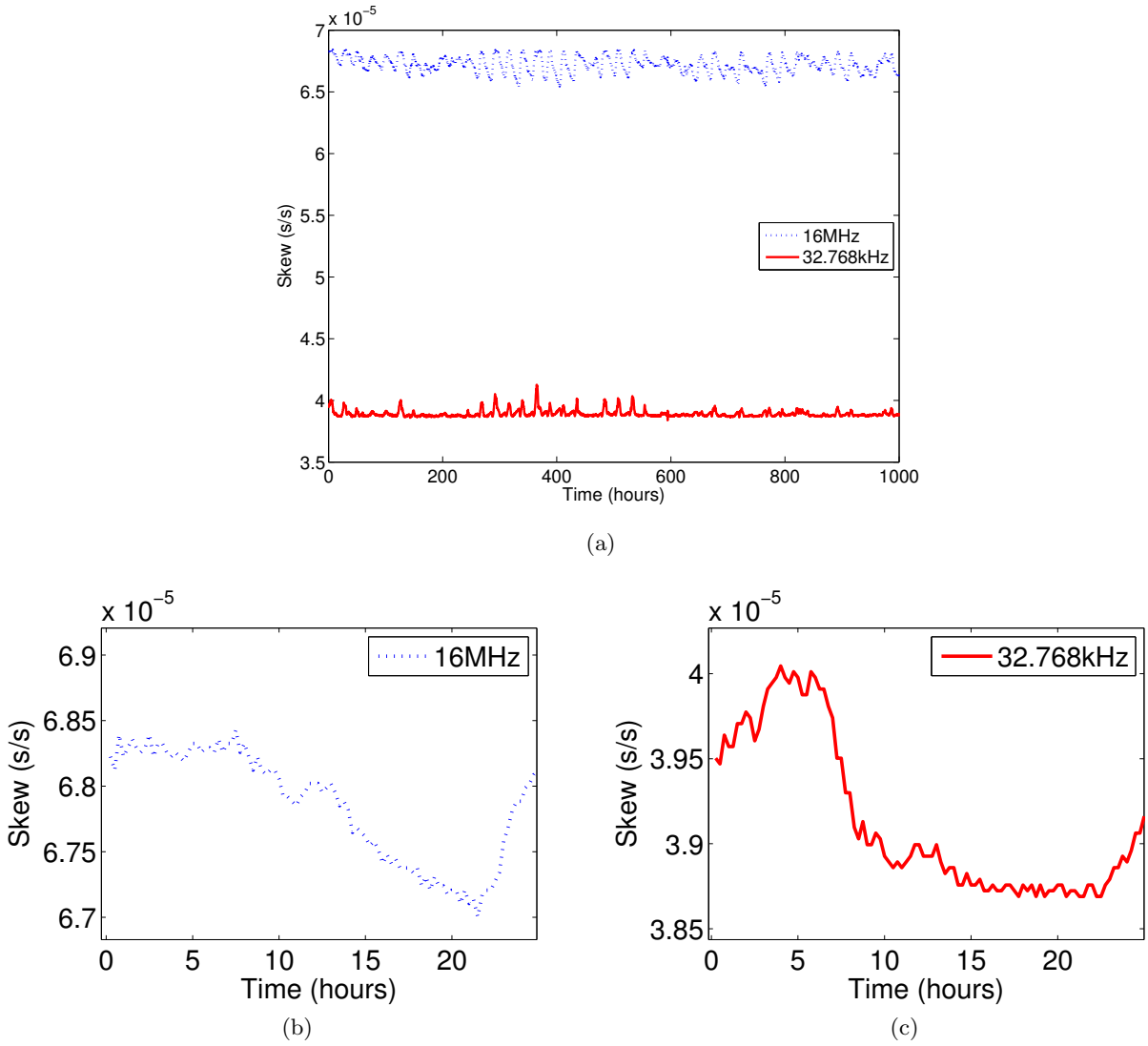


Figure 6.1: Measurement of Time-Varying Clock Skews

environments. It is expected that the skews would vary severely in the real environment such as lack of energy, large temperature variation. Therefore we need a model which can reflect a time-varying characteristic instead of the two models in (6.5) and (6.6). In the following, we give a model for the random clock skew starting from the phase noise.

Phase noise in oscillators has different representations. Here we consider a simple way to model it through jitter. It is natural to think of it as a noisy random offset in the timing of events. If the unperturbed oscillator output is $s(t)$, the jitter perturbed output is $s(t + \phi(t)/2\pi f_o)$, where $\phi(t)$ is random and f_o is the center frequency of the oscillator. Clearly the jitter $\phi(t)$ affects the frequency of the oscillator $\frac{d\phi(t)}{dt}\frac{1}{2\pi f_o}$ and thus causes clocks to have random offset and skew [**?**]. In general, phase noise is not stationary but only cyclostationary.

To model the time-varying clock skew as a random process, we assume clock skew is a random process with zero mean and a small perturbation around the mean. This assumption has been observed by some previous works (e.g., [57]) which adopt constant skew. Here we assume the smoothness (order

of auto-regression (AR) model) of the clock skew is P, which is more general than the previous works [57], [79]. This means that the clock skew satisfies the auto-regressive relation as:

$$\alpha[n] = \sum_{i=1}^{P} c_i \alpha[n-i] + \eta[n], \tag{6.7}$$

where c_i's are AR coefficients, $\eta[n]$ is model noise with zero mean and σ_η^2 variance. Usually we model $\eta[n]$ as Gaussian noise because in general, the phase noise derivative $\Delta\phi(t)$ is unbounded, but the frequency drift is focused within a certain range (which is usually specified by the oscillator manufacturer)[1]. This model is general and practical. It subsumes the two extreme cases in (6.5) and (6.6) as special cases. It also quantifies the drifting of the clock frequency, captures the main variation of the skew and also takes into account the randomness.

To apply this model, the parameters c_i's need to be determined. There are two ways to estimate c_i's. One is to estimate these coefficients based on the statistical properties of $\alpha[n]$. Define the auto-correlation function of $\alpha(t)$ as $r_\alpha(\tau) = E\{\alpha(t)\alpha(t+\tau)\}$, and then the AR($P$) coefficients c_i's can be derived as:

$$\begin{bmatrix} r_\alpha(P\tau) \\ r_\alpha((P-1)\tau) \\ \vdots \\ r_\alpha(\tau) \end{bmatrix} = \begin{bmatrix} r_\alpha(0) & \dots & r_\alpha((1-P)\tau) \\ r_\alpha(\tau) & \dots & r_\alpha((2-P)\tau) \\ \vdots & \vdots & \vdots \\ r_\alpha((P-1)\tau) & \dots & r_\alpha(0) \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_P \end{bmatrix}. \tag{6.8}$$

It is clear that as time goes on, the auto-correlation of the clock skews becomes weaker. In [80], the auto-correlation function is modeled as a decaying exponential as:

$$r_\alpha(\tau) = \sigma_\alpha^2 \rho^{\frac{\tau}{\nu}}, \tag{6.9}$$

where ν denotes the normalizer to model different decaying rates, and ρ is a positive number close to 1. Here in this chapter, we also adopt this exponential decaying model for the autocorrelation function. To estimate the parameters c_i's of AR(P) model, we need to estimate the auto-correlation function and variance of $\alpha[n]$. Given the clock observations $\theta[n]$ in (6.3), one can obtain samples of the clock skew $\alpha[n]$'s using an independent skew model in (6.6). Thus, the autocorrelation $r_\alpha(\tau_0)$ and the variance can be estimated using sample means. Once we obtain the auto-correlation, we can find its parameters by setting $\nu = \tau_0$ and solving for ρ. Then we can use this auto-correlation to estimate c_i's for the desired sampling period τ. Theoretically $\alpha(t)$ is non-stationary, and thus coefficients c_i's may change along with time. However, c_i's change quite slowly relative to the clock offset and thus we can still take them as quasi-stationary.

The other way is to use some training data. Suppose that we collect T observations where $T > P$ as the model fitting data, and then (6.7) can be rewritten as:

$$\begin{bmatrix} \alpha[P+1] \\ \alpha[P+2] \\ \vdots \\ \alpha[T] \end{bmatrix} = \begin{bmatrix} \alpha[P] & \dots & \alpha[1] \\ \alpha[P+1] & \dots & \alpha[2] \\ \vdots & \vdots & \vdots \\ \alpha[T-1] & \dots & \alpha[T-P] \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_P \end{bmatrix} + \begin{bmatrix} \eta[P+1] \\ \eta[P+2] \\ \vdots \\ \eta[T] \end{bmatrix}. \tag{6.10}$$

[1]Note that this assumption is not required for the derivation of the following Kalman filter.

Let $\mathbf{z}$, $\mathbf{Z}$, $\mathbf{c}$ and $\boldsymbol{\eta}$ denote the four column vectors/matrix in (6.10) for simplicity. Then performing QR-decomposition on $\mathbf{Z}$, we obtain an upper-triangular matrix $\mathbf{R}$. We can rewrite (6.10) as $\mathbf{Rc} = \mathbf{d}$ where $\mathbf{d} = \mathbf{Q}^T\mathbf{z}$ ($\mathbf{Q}$ is an orthogonal matrix). $\mathbf{Rc} = \mathbf{d}$ is solved by backward substitution. In this way, given a model order P, we can find the AR coefficients in (6.10) using least squares approach.

6.3.4 Information Criteria for Model Order Selection

To model the time-varying clock skews as AR processes, the selection of model order P in (6.7) is an important step. As the model order increases, the model becomes more accurate to the measurements; however, if model order is too high, there may be other kinds of problems such as increasing complexity while the fitting performance does not improve any more. Following principle of Occam's razor, we adopt information criteria to select a proper model order.

The widely adopted information criterion is Akaike Information Criterion (AIC) [81]. When using finite dimensional AR models, the AIC provides an asymptotically efficient solution under quadratic loss function to select the best fit to the data. AIC finds an appropriate order of a model. It uses a maximum likelihood function of the model while adding a modeling penalty function which is a term to represent complexity generated as a modeling order increases. The modeling penalty function in AIC is the number of estimate parameters in the model (see [81–84]). AIC is defined as:

$$\text{AIC}(P) = -2(\text{maximized log likelihood of the model}) + 2(\text{the number of estimated parameters in the model}). \tag{6.11}$$

It can be simplified to (6.12) [81]:

$$\text{AIC}(P) = T\log(2\pi\hat{\sigma}_P^2) + 2P, \tag{6.12}$$

where $\hat{\sigma}_P^2$ is an estimate of the variance of the modeling error $\eta[n]$in (6.7) as:

$$\hat{\sigma}_P^2 = \frac{1}{T-P}\sum_{n=P+1}^{T}(\alpha[n] - \sum_{i=1}^{P}\hat{c}_i\alpha[n-i])^2, \tag{6.13}$$

with $\hat{c}_i$ denoting the estimate of the coefficients from (6.8) or (6.10).

Another often used information criterion is called Minimum Description Length (MDL). This criterion also consists of the maximized log-likelihood function as AIC, and uses $(\log T)P$ as a penalty function shown as:

$$\text{MDL}(P) = T\log(2\pi\hat{\sigma}_P^2) + (\log T)P. \tag{6.14}$$

By multiplying P by log$T$ in the penalty function, the penalty term becomes larger as the number of observations increases. It says that MDL tries to find the smaller number of parameters compared to AIC when having large number of observations.

The third information criterion is AIC_c which is modified based on AIC in (6.12) and gives a better modeling order when the number of samples is small. AIC may perform poorly when the number of parameters in the model under consideration is a substantial fraction of the sample size. AIC_c adds

$\frac{2P(P+1)}{T-P-1}$ to AIC when a sample size is small as:

$$\mathrm{AIC}_c(P) = T\log(2\pi\hat{\sigma}_P^2) + \frac{2T}{T-P-1}P. \tag{6.15}$$

The optimal order of AR processes can be decided when the results of these information criteria are minimized. Using this optimal order and the estimates of AR parameters, the time-varying clock skew model in (6.7) is generated.

Given the recursive observation model in (6.4) and the AR model in (6.7), we are finally prepared to construct Kalman filter as a frame work to track the variation of the clock skew and synchronize the clocks. In the following section, we will introduce how to apply Kalman filter for time synchronization.

6.4 Clock Skew and Offset Tracking

Ideally, it is possible that clock behaviors are estimated accurately if an exact model is derived. However, in real world, clocks are affected by environmental factors such as temperature variations, vibration and humidity. Moreover, resource-constrained networks, for example, WSNs usually consist of inexpensive devices which have unstable oscillators, vulnerable to interferences. Noise, which is generated by environmental interferences, is added to clock information as observation noise when clock reading and distorts clock information. This prevents even an exact clock model from tracking clock behaviors exactly. Hence, preprocessing is necessary to approximate the original clock information when noisy observation is delivered. In this section, we design Kalman filters to track the time-varying clock skew and offset based on the proposed clock skew and clock offset models.

6.4.1 Skew and Offset Estimation with a Kalman Filter

Suppose that the sampling rate is fixed, i.e., uniform sampling with $\tau[n] = \tau_0$. The sampling period τ_0 is known and the coefficients c_i's of the AR(P) model in (6.7) is estimated. Let $\tilde{\theta}[n]$ denote the true clock offset (i.e., $\tilde{\theta}[n] = \sum_{k=1}^{n} \alpha[k]\tau[k] + \theta_0$). It is clear that $\tilde{\theta}[n] = \tilde{\theta}[n-1] + \alpha[n]\tau_0$. We can define an extended state equation as:

$$\boldsymbol{x}[n] = \boldsymbol{A}\boldsymbol{x}[n-1] + \boldsymbol{u}[n], \tag{6.16}$$

where $\boldsymbol{x}[n] = \left[\, \tilde{\theta}[n] \;\; \alpha[n] \;\; \ldots \;\; \alpha[n-P+1] \,\right]^T$,

$$\boldsymbol{A} = \begin{bmatrix} 1 & \tau_0 & 0 & \ldots & 0 \\ 0 & \hat{c}_1 & \hat{c}_2 & \ldots & \hat{c}_P \\ 0 & 1 & 0 & \ldots & 0 \\ 0 & 0 & 1 & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \ldots & 0 \end{bmatrix}, \; \boldsymbol{u}[n] = \begin{bmatrix} 0 \\ \eta[n] \\ 0 \\ \vdots \\ 0 \end{bmatrix},$$

where $\boldsymbol{A}$ is a $(P+1)\times(P+1)$ transition matrix and $\boldsymbol{u}[n]$ is a $(P+1)\times 1$ driving noise vector. The observation equation is defined as:

$$\theta[n] = \tilde{\theta}[n] + v[n] = \boldsymbol{b}^T\boldsymbol{x}[n] + v[n], \tag{6.17}$$

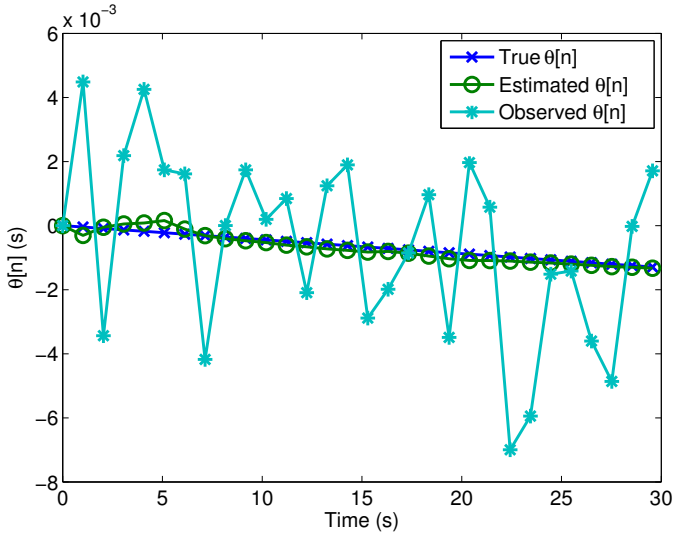


Figure 6.2: Example: Tracking Clock Offset

where $\boldsymbol{b}^T = [\,1\ 0\ \ldots\ 0\,]$, which is a $1 \times (P+1)$ vector, and $v[n]$ is observation noise. In this case, the Kalman filter design is summarized as follows (cf. [85, Chp. 13]).

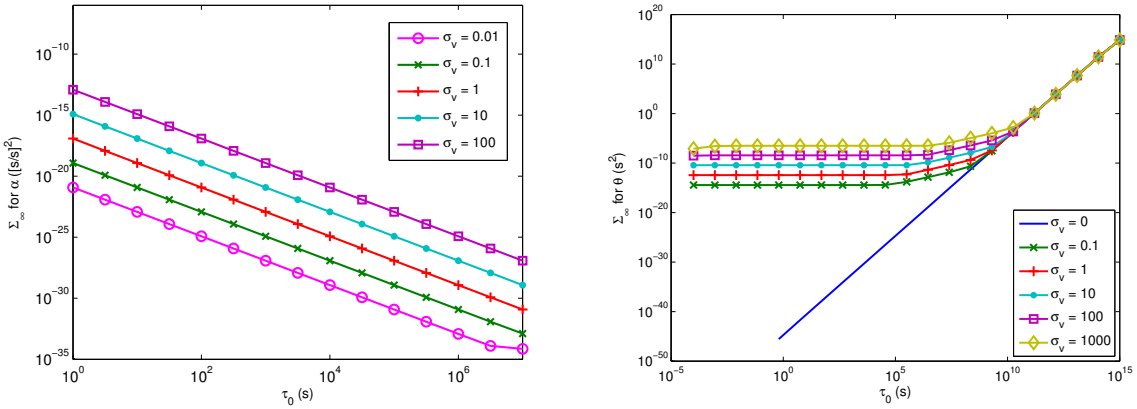
$$\begin{aligned}
\text{Update}: &\hat{\boldsymbol{x}}[n] = \boldsymbol{A}\hat{\boldsymbol{x}}[n-1] \\
&\quad + \boldsymbol{G}[n](\theta[n] - \boldsymbol{b}^T\boldsymbol{A}\hat{\boldsymbol{x}}[n-1]) && (6.18)\\
\text{MSE}: &\boldsymbol{\Sigma}[n] = \boldsymbol{A}\boldsymbol{M}[n-1]\boldsymbol{A}^T + \boldsymbol{C}_u && (6.19)\\
&\boldsymbol{M}[n] = (\boldsymbol{I} - \boldsymbol{G}[n]\boldsymbol{b}^T)\boldsymbol{\Sigma}[n] && (6.20)\\
\text{Kalman Gain}: &\boldsymbol{G}[n] = \boldsymbol{\Sigma}[n]\boldsymbol{b}\left(\sigma_v^2 + \boldsymbol{b}^T\boldsymbol{\Sigma}[n]\boldsymbol{b}\right)^{-1}, && (6.21)
\end{aligned}$$

where $\boldsymbol{\Sigma}[n]$ is the prediction MSE of the estimate when the current observation is not considered, $\boldsymbol{A}\hat{\boldsymbol{x}}[n-1]$. $\hat{\boldsymbol{x}}[n]$ is the estimate of the offset and skew state at the n^{th} sample, $\boldsymbol{M}[n]$ is the minimum mean-square error (MMSE) of the estimate, and $\boldsymbol{G}[n]$ is the so-called Kalman gain. σ_v^2 is the observation noise variance, and $\boldsymbol{C}_u$ is the covariance matrix of $\boldsymbol{u}[n]$ when σ_η^2 is the variance of the driving noise in the state equation. The recursion of the Kalman filter is initialized by

$$\hat{\boldsymbol{x}}[0] = \begin{bmatrix} \mathrm{E}\{\theta[n]\} \\ \mathrm{E}\{\bar{\boldsymbol{\alpha}}[n]\} \end{bmatrix}, \quad \boldsymbol{M}[0] = \begin{bmatrix} \sigma_v^2 & 0 \\ 0 & \sigma_\alpha^2 \boldsymbol{I}_P \end{bmatrix},$$

where $E\{\cdot\}$ denotes the statistical expectation, and $\bar{\boldsymbol{\alpha}}[n]$ is a $P \times 1$ vector consisting of $\alpha[n]$. σ_α^2 is the variance of $\alpha[n]$, and $\boldsymbol{I}_P$ is a $P \times P$ identity matrix. Since Kalman filter does not strongly depend on the initial conditions, the statistical mean and variance can be replaced by sample mean and sample variance. For example, $\hat{\alpha}[0]$ can be initialized as $(\theta[1] - \theta[0])/\tau_0$.

An example profile of this algorithm is shown in Fig. 6.2. The observation noise variance is $\sigma_v^2 = 10^{-5}$ s^2, and the parameter ρ in (6.9) is chosen as $1 - 2 \cdot 10^{-6}$ with $\nu = 1$ hour. For simplicity, we assume an AR(1) model for clock skew in (6.7) and recursive observation model in (6.4). Fig. 6.2



(a) Σ_∞ for α (b) Σ_∞ for θ

Figure 6.3: Example: Steady State Prediction MSEs

shows the offset estimated by our algorithm from (6.18) - (6.21) ("Estimated $\theta[n]$"), the true offset $\tilde{\theta}[n]$ ("True $\theta[n]$"), and the offset from observation ("Observed $\theta[n]$"). Note that even though the observed offset is dominated by the observation noise, the Kalman filter is able to extract the true value with only small deviations.

Fig. 6.3 shows the effects of sampling frequency and measurement noise. As shown in Fig. 6.3a, while the sampling period τ_0 increases, the MSE for the estimate of $\alpha[n]$ decreases. Initially, this claim may sound counter-intuitive since one may think the faster the sampler is, the better the estimator performs. However, increasing τ_0 reduces the variance of the effective observation. If we take the observation equation in (6.4) we can solve for $\alpha[n]$ and write:

$$\alpha[n] = \frac{\tilde{\theta}[n] - \tilde{\theta}[n-1] + v[n]}{\tau_0} = \frac{\tilde{\theta}[n] - \tilde{\theta}[n-1]}{\tau_0} + \tilde{v}[n], \tag{6.22}$$

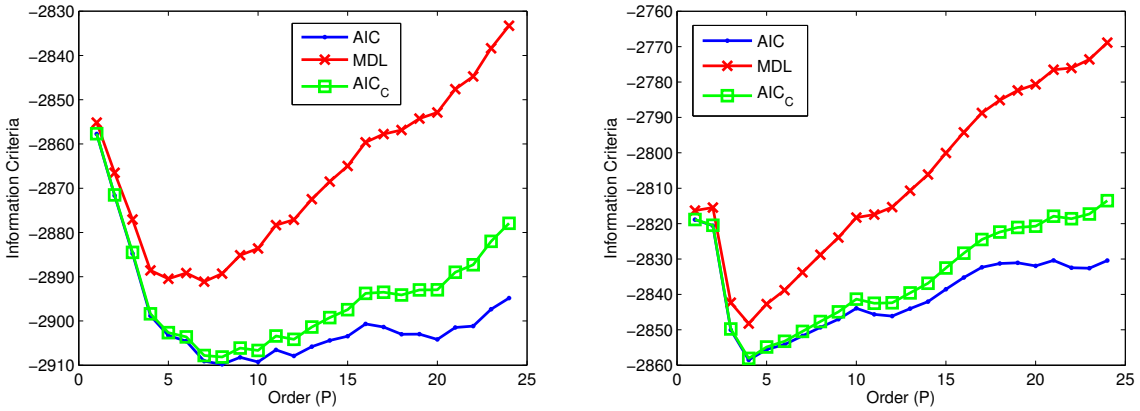
where $\tilde{v}[n] = \frac{v[n]}{\tau_0}$ has variance $\frac{\sigma_v^2}{\tau_0^2}$. This $\tilde{v}[n]$ is the effective noise when used as an estimate of $\alpha[n]$. This means the reduced MSE for larger sampling period is purely due to reduced effective observation noise. Note from Fig. 6.3b that this decrease in MSE for $\alpha[n]$ does not appear to significantly affect the offset estimate. The observation equation for offset θ in this model has noise with variance σ_v^2 which does not depend on τ_0. From this it is clear why the MSE is relatively constant here.

6.5 Experiment and Model Validation

In this section, we verify the proposed clock models and the Kalman filtering method with measurements of real low-cost clocks empirically, and show their performance.

6.5.1 Clock Skew and Offset Measurement

In order to ensure applicable results we use a low-powered micro-controller platform similar to that deployed in sensor networks. Like the Mica2 mote [86], the device we tested uses an Atmel ATmega128L



(a) 32.768kHz Clock

(b) 16MHz Clock

Figure 6.4: Model Order Selection

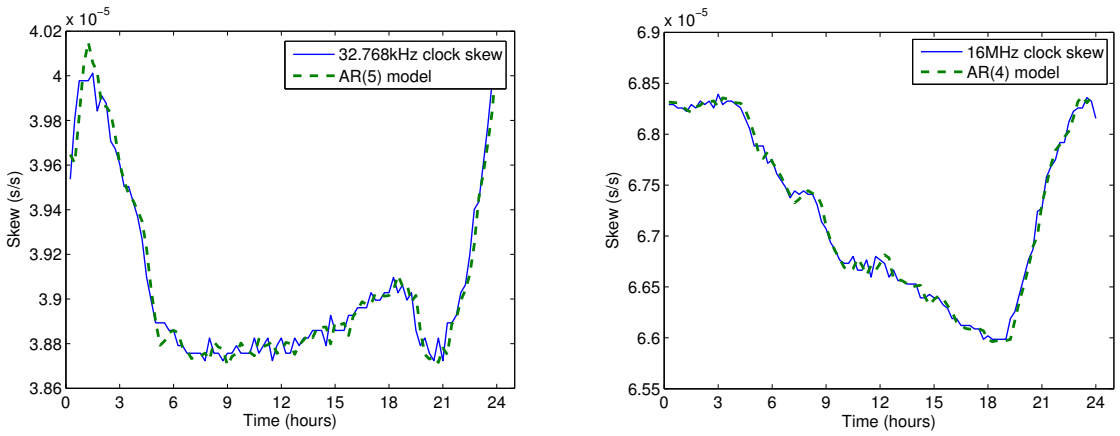
processor with a 32.768kHz crystal oscillator. Our hardware platform differs in that it uses a 16MHz primary crystal oscillator to drive the processor instead of the 8MHz present on the Mica2 mote, and lacks the RF interface.

The device was programmed to maintain two counters representing the time in $1/32768^{\text{th}}$s of a second. One counter was driven by the 16MHz external crystal oscillator and the other counter was driven by the external 32.768kHz crystal oscillator. The device was connected to a PC over an RS232 serial interface running at 38400 baud and was programmed to respond to timestamp queries. The timestamp queries consisted of the PC sending a single byte as a synchronization point followed by a 4-byte timestamp. The device would then reply by sending a 4-byte timestamp for each of the counters in response. Upon reception of the single byte, the device disables interrupts, copies the current time value from both counters to temporary registers, enables interrupts, and transmits the timestamps a byte at a time.

Timestamps were recorded on a 2.4GHz Athlon 64 Dual Core PC running Ubuntu 8.04. A program was run on the PC to send timestamp requests and record the replies in a binary file. In order to reduce latency on the PC, the program used the "mlockall" call, preventing the program from being swapped to disk, to avoid paging delays and enabled real-time scheduling using the SCHED_FIFO scheduler. Additionally the serial port involved was set to low-latency mode using the setserial program. CPU frequency scaling and NTP were disabled on the PC to eliminate their effects on measurements.

Time was estimated on the PC using the "gettimeofday" call and converting the time to $1/32768^{\text{th}}$s of a second. The device was placed under direct sunlight in the room during the test to better simulate outdoor environment of sensors. Timestamps were collected using this setup every second for over 1.5 months.

We calculate the skew between samples separated by several minutes over the entire dataset. This is equivalent to applying a moving average filter of the same width to the skew calculated from adjacent samples. This technique removes the effects of the quantization noise, but larger filter widths will remove some dynamic components. The instantaneous offset $\theta[n]$ was obtained as $\theta[n] =$



(a) 32.768kHz Clock Skew

(b) 16MHz Clock Skew

Figure 6.5: Comparison of an AR Model and Measurement of Clock Skew

$t_{\text{PC}}[n] - t_{\text{device}}[n]$, where $t_{\text{PC}}[n]$ is the instantaneous time of PC and $t_{\text{device}}[n]$ is that from counters of the device, every 900 seconds to reduce the effects of noise for our simulation. The skew $\alpha[n]$ was obtained as $\alpha[n] = \frac{\theta[n+1]-\theta[n]}{t_{\text{PC}}[n+1]-t_{\text{PC}}[n]}$ for the same interval as the offset. The measured results are shown in Fig. 6.1. Over the course of the measurement, the skew was around 40ppm for the 32.768kHz clock and around 70ppm for the 16MHz clock. Additionally the skews of both clocks vary as time and the variation is as much as 3ppm over this course. Moreover, the accumulated offset was about 150(s) for the 32.768kHz clock and about 250(s) for the 16MHz clock.

6.5.2 Model Order Selection and AR(P) Model

Since our clock tracking method highly depends on the AR(P) model, we first use the real measurements to validate the AR(P) model and select a reasonable model order P. To find an appropriate order of an AR model for each clock skew, information criteria which are introduced in Section 6.3.4 are employed. Here we choose one-day data for model fitting, i.e., $T = 96$ samples based on the observation from Fig. 6.1. For each information criterion, we use the one day data to calculate a penalty function to find a fitting order of the AR model. As Fig. 6.4 shows, the resulting values of AIC, MDL and AIC$_c$ have similar patterns. The optimum orders of model based AIC, MDL, and AIC$_c$ are 8, 7, and 8, respectively for 32.768kHz clock (see Fig. 6.4a) and 4 for 16MHz clock (see Fig. 6.4b). We also observe that MDL criterion outperforms AIC and AIC$_c$ by giving sharper cost functions.

Fig. 6.5 illustrates that the derived AR models match each clock behavior with the number of parameters less than 5. We do not choose an optimum from information criteria as the model order of the 32.768kHz clock. Since clock models in resource-limited networks need to be as less complex as possible, we choose the order at which a downward tendency in the information criteria curves decreases. The optimum from information criteria is employed as the model order for 16MHz clock. The AR(5) model of the 32.768kHz is applied to a Kalman filter to show that the proposed method tracks real clock behaviors in the following.

6.5.3 Clock Skew and Offset Tracking

Here we estimate the real clock skews and offsets based on the Kalman filter approach developed in Section 6.4 and evaluate our method in various cases.

Tracking Performance

Suppose that a sampling period τ_0 is fixed as 900 seconds, which is a known value. The AR model derived in Section 6.5.1 is used as a clock skew model. When $\tilde{\theta}[n]$ is the true clock offset and $\alpha[n]$ is the true clock skew from the measurements (i.e., $\alpha[n] = \frac{\tilde{\theta}[n]-\tilde{\theta}[n-1]}{\tau_0}$), the state equation is defined as (6.16), where $\boldsymbol{x}[n] = \left[\, \tilde{\theta}[n] \;\; \alpha[n] \;\; \alpha[n-1] \;\; \alpha[n-2] \;\; \alpha[n-3] \;\; \alpha[n-4] \,\right]^T$,

$$\boldsymbol{A} = \begin{bmatrix} 1 & \tau_0 & 0 & 0 & 0 & 0 \\ 0 & \hat{c}_1 & \hat{c}_2 & \hat{c}_3 & \hat{c}_4 & \hat{c}_5 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}, \; \boldsymbol{u}[n] = \begin{bmatrix} 0 \\ \eta[n] \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix},$$

where $\hat{c}_i$'s are the estimates of AR coefficients. The observation equation is defined as (6.17), where $\boldsymbol{b}^T = \left[\, 1 \; 0 \; 0 \; 0 \; 0 \; 0 \,\right]$ and $\theta[n]$ is the observation of the clock offset and $v[n]$ is an observation noise which represents any kinds of uncertainties including network delay and measurement errors which can be added to the true data when the clock offset is observed.

With these conditions, we design Kalman filter as (6.18) - (6.21) where $\hat{\boldsymbol{x}}[n]$ is the estimate of $\boldsymbol{x}[n]$ and $\boldsymbol{\Sigma}[n]$ is the prediction MSE of the estimate when the current observation is not considered, $\boldsymbol{A}\hat{\boldsymbol{x}}[n-1]$. $\boldsymbol{C}_u$ is the covariance matrix of $\boldsymbol{u}[n]$ when $\eta[n]$ is the modeling error and its variance is $3.91502 \cdot 10^{-15}(s^2)$ (calculated based on the measurement as in (6.13)). The recursion of the Kalman filter is initialized by

$$\hat{\boldsymbol{x}}[0] = \begin{bmatrix} \tilde{\theta}[1] \\ \bar{\boldsymbol{\alpha}}[1] \end{bmatrix}, \quad \boldsymbol{M}[0] = \begin{bmatrix} \sigma_v^2 & 0 \\ 0 & \sigma_\alpha^2 \boldsymbol{I}_5 \end{bmatrix},$$

where $\bar{\boldsymbol{\alpha}}[1]^T = \left[\, \alpha[1] \; \alpha[1] \; \alpha[1] \; \alpha[1] \; \alpha[1] \,\right]$, σ_v^2 is the variance of the measurement noise of clock offsets, and σ_α^2 is the variance of clock skew, $\alpha[n]$, which is $1.29446 \cdot 10^{-13}(s^2)$. The estimates of AR coefficients, which are components of $\boldsymbol{A}$, are obtained from (6.10).

Fig. 6.6 illustrates that the estimated clock skews are close to the true clock skews even when observed skews are fluctuating due to noise. Here we can see that Signal-to-noise ratio (SNR) of clock offset is set to -20dB by setting the standard deviation of observation noise of clock offset, σ_v, as $3 \cdot 10^{-4}(s)$. Even though the variances of clock skews induced by the observation noise of clock offsets are much greater than the variance of the true clock skews, the designed Kalman filter tracks clock skews well. This means the derived skew model is accurate to track the real measured clock which is unstable and time-varying.

To validate our Kalman filtering method, more simulations are conducted and the prediction MSE is used as a performance metric for evaluation. First, we analyze the effects of the varying sampling rate on clock estimation performance. Since clock synchronization consumes resources such as power, hardware and processing time, even though our approach is simple, some resource-constrained networks

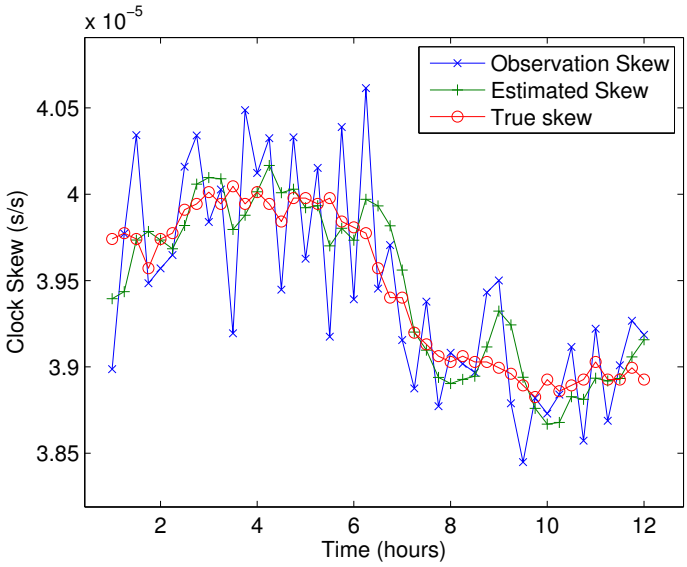
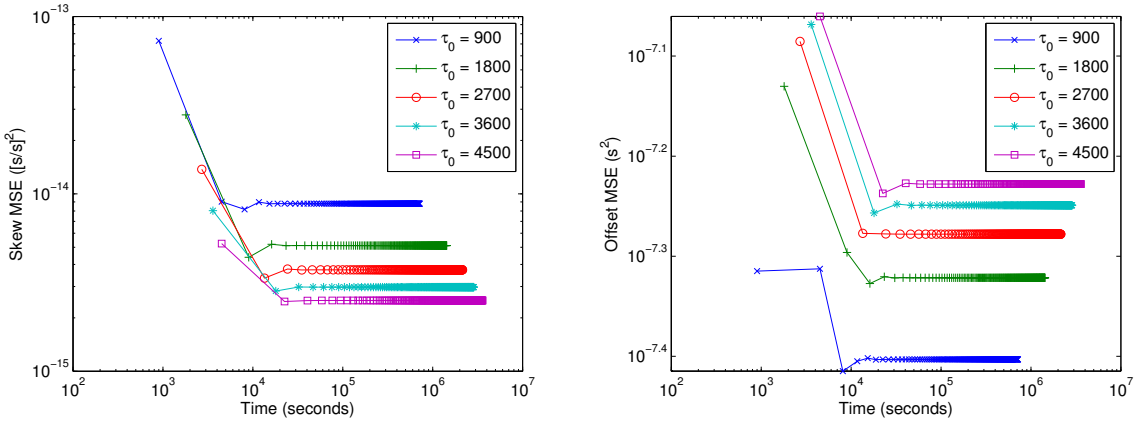


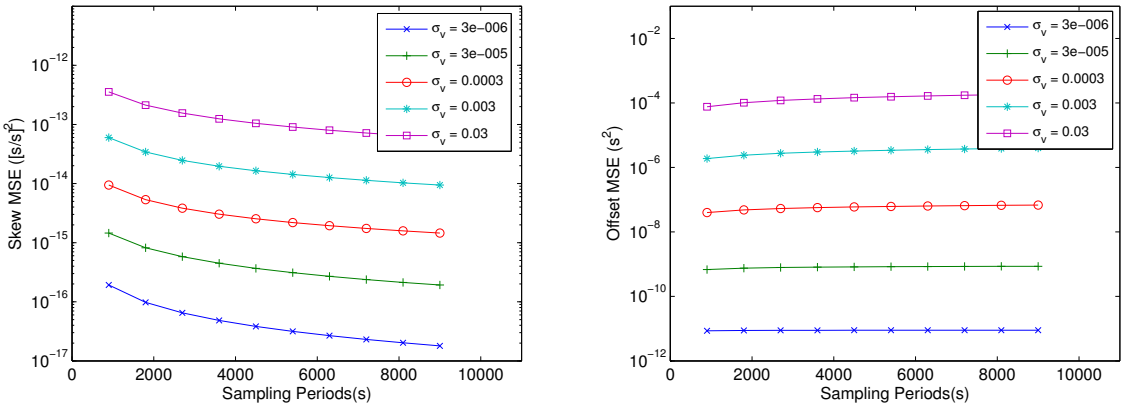
Figure 6.6: Tracking Clock Skew using Kalman Filter



(a) Skew MSE (b) Offset MSE

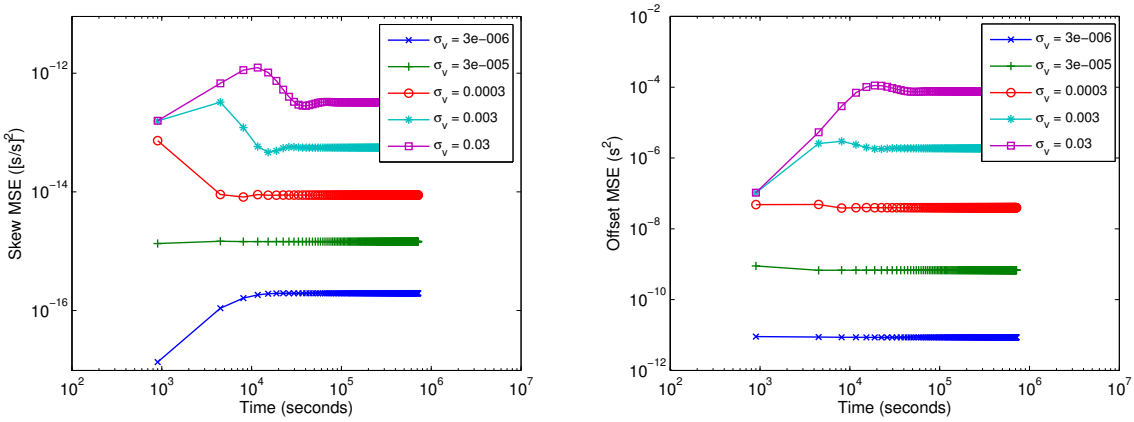
Figure 6.7: MSEs for Various Sampling Rates

need to synchronize clocks infrequently to conserve their resources. Fig. 6.7 depicts the performance of estimating clock skews and clock offsets as synchronization period varies. The simulation was continued for 800 synchronization periods with several different sampling periods and these curves represent the prediction MSEs of skew and offset every 4 samples. The MSE of the skew in Fig. 6.7a becomes smaller when sampling rate is lower. This is because that longer sampling period cannot capture the small and fast variations of the skew and thus the modeling error is smaller. This can also be explained by the calculation of skew from the offset observations as $\alpha[n] = \frac{\theta[n]-\theta[n-1]}{\tau_0}$. When τ_0 is large, the noise variance in observed clock skew is reduced to $\frac{\sigma_v^2}{\tau_0^2}$. For the offset in Fig. 6.7b, the MSEs change little, around $10^{-7.3}(s^2)$, even as sampling period varies. It means the performance of tracking offset hardly diminish when sampling period becomes longer. Fig. 6.7 also shows that both the MSEs of clock skews and offsets converge to steady state after several samples. As the sampling



(a) Skew MSE vs. Sampling Periods (b) Offset MSE vs. Sampling Periods

Figure 6.8: Steady State MSE vs. Sampling Periods



(a) Skew MSE (b) Offset MSE

Figure 6.9: MSEs for Various Observation Noise Variances

period τ_0 increases, the prediction MSE approaches a certain level in fewer samples. However, since greater τ_0 implies a larger sampling period, it actually has longer convergence time. The performance shown is only at sampling instants, where new data has just arrived, rather than at arbitrary time instants. The performance will therefore be better than the true time-averaged performance.

Fig. 6.8a and Fig. 6.8b further clarify the relation between sampling rate and MSEs as showing the steady state MSEs. The simulations were run 200 times and averaged for each point. The skew MSE lessens as sampling rate decreases while the offset MSE is nearly same. Both of them are strongly influenced by observation noise variance than sampling period. In both cases, when standard deviation of observation noise becomes ten times greater, MSEs increase almost ten times. This result is compatible with the previous section (see Fig. 6.3). It shows an example which uses the simple AR(1) model of the clock skew. We have shown that our modeling and tracking methods are quite robust to different sampling rates.

As the variance of observation noise increases, the prediction MSEs of the clock skew and the clock

Table 6.1: Performance Comparison

	Skew MSE $([s/s]^2)$	Offset MSE (s^2)
AR(5) Skew Model	$1.2395 * 10^{-14}$	$4.5397 * 10^{-8}$
AR(1) Skew Model	$1.7641 * 10^{-14}$	$6.7588 * 10^{-8}$
Constant Skew Model	$8.2979 * 10^{-14}$	$4.3721 * 10^{-3}$

offset increase and the MSE curves become stable as time goes on as shown in Fig. 6.9a and Fig. 6.9b. To obtain MSE curves, the tracking algorithm was run for 800 synchronization periods with several different noise variances. MSEs of the estimated clock skews, $\hat{\alpha}[n]$, are less than $10^{-12}([s/s]^2)$ even when σ_v is as high as $0.03(s)$, and those of the estimated clock offsets, $\hat{\theta}[n]$, are less than $10^{-4}(s^2)$ at the same condition. Even if the variance of observation noise is extremely high, the prediction MSE of our Kalman filter curves converge in several samples.

Comparison

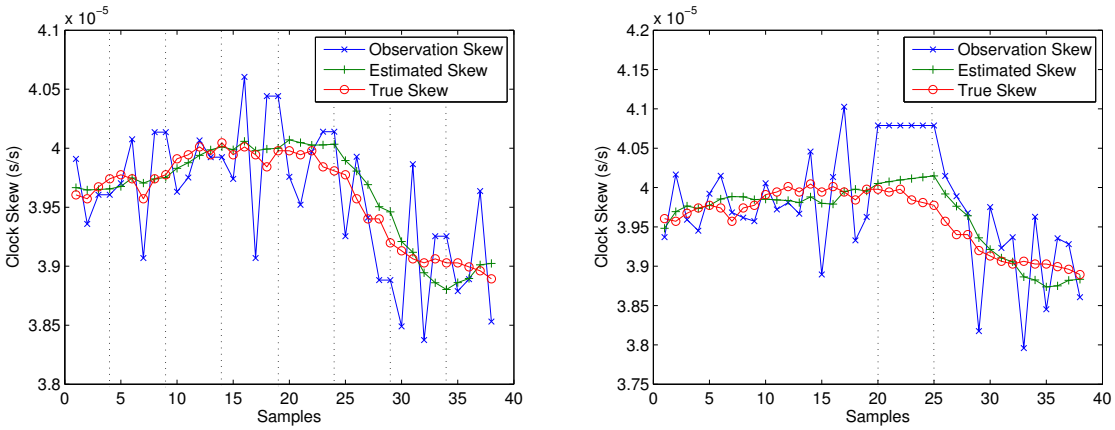
In this section, to evaluate our clock models, we borrow two other skew models and compare them. Simulation was run 200 times for 4000 synchronization periods when τ_0 is $900(s)$ and σ_v is $3 \cdot 10^{-4}(s)$. Mean square error calculated by (6.23) and (6.24) is used as a performance metric for evaluation:

$$\mathrm{MSE}_{offset}[n] = \frac{\sum_{i=1}^{n} (\hat{\theta}[i] - \tilde{\theta}[i])^2}{n}, \tag{6.23}$$

$$\mathrm{MSE}_{skew}[n] = \frac{\sum_{i=1}^{n} (\hat{\alpha}[i] - \tilde{\alpha}[i])^2}{n}. \tag{6.24}$$

First, an AR(1) skew model is generated and used as a skew model instead of the AR(5) model while (6.4) is used as the offset model as before. MSEs are calculated with the result to track the varying clock skew and offset using our method with the newly generated clock skew model. As shown in Table 6.1, both of the skew MSE and the offset MSE are higher than those with the AR(5) skew model which is the clock skew model with the optimal number of order by information criteria. This explains that to derive the model of varying skews using the optimal order enhances the tracking performance.

Another skew model for performance comparison is a constant skew model in (6.5). When the clock skew is constant, the clock offset represents as a linear model. MSEs in Table 6.1 are calculated with a constant skew and a linearly increasing offset. Both MSEs are greater than those of the AR(1) and AR(5) model. Especially the offset MSE is 10^5 times worse compared to that of AR(1) and AR(5). This shows the importance to use a time-varying model for low-precision clock skews. Since observation noise is not added for simulations of a constant skew model while σ_v is $3 \cdot 10^{-4}(s)$ for simulations of AR(1) and AR(5), performance of a constant skew model will be worse if observation noise is considered. Constant skew model is not suitable for varying skews.



(a) Uniform Missing Data

(b) Consecutive Missing Data

Figure 6.10: Tracking Clock Skew with Missing Data

6.6 Tracking with Corrupted Data

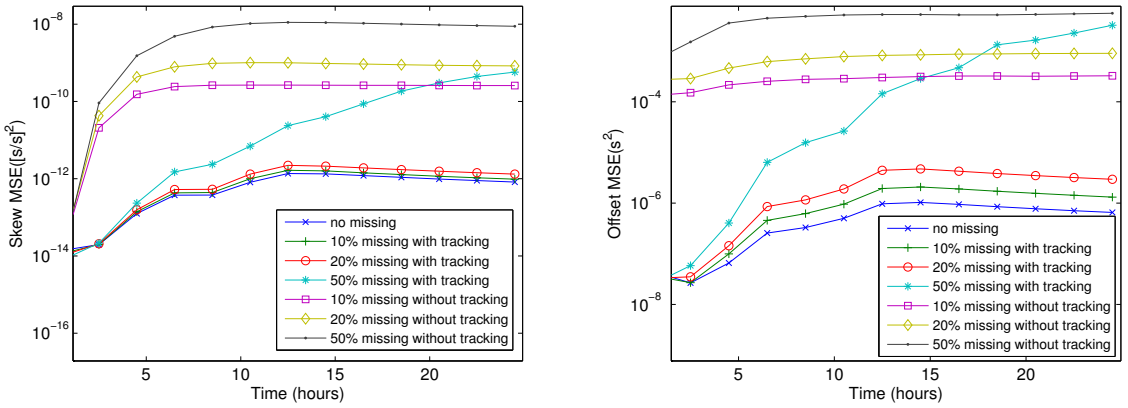
In sensor networks, the links are unreliable and interfered and packets which contain timestamps may get lost or collided. In this case, the received timestamps are not uniform and/or corrupted. That causes missing and corrupted data. In this section, we show our tracking method with corrupted data.

6.6.1 Tracking with Missing Data

We test the tracking performance with three types of missing data: (i) uniform loss; (ii) consecutive loss; and (iii) random loss. Again Kalman filter in Section 6.4 is adopted and set the same as in Section 6.5.3.

When missing data, observation is not updated and stays the same value as the previous one. Since we do not have a new observation, we can not estimate clock skew and clock offset using the introduced Kalman filtering method. However, the previous estimates instead of observation facilitate to find new estimates of clock skew and offset. Fig. 6.10a shows the tracking performance of the clock skew when one out of every five observations is dropped. Vertical dot lines indicate the points of missing data. Clock skew and clock offset can still be tracked closely with true clock skews based on the AR model derived in Section 6.5.1 when the clock information is absent.

Next, let us suppose packets do not arrive at destination node for a while due to a mechanical problem of devices, or channel congestion. Fig. 6.10b shows an example of tracking clock skew when six consecutive data are missing. Even if several consecutive data packets are missing, the proposed method follows the true skew closely. Since the observation is not updated, clock skew and offset are tracked only with estimates based on the model during the missing period. The estimated skew is close to the true skew even for consecutive missing. However, the gap between the estimate of clock skew and the true skew becomes larger as data missing lasts. It means our approach tracks the unreliable clock even when a system misses clock information for a while; however, if packets which include clock information are missed consecutively for a long period, our method cannot track clocks. Because AR



(a) Skew MSE with Random Missing Data

(b) Offset MSE with Random Missing Data

Figure 6.11: MSEs with Random Missing Data

model is a statistical forecasting model in which future values are computed only on the basis of past values of a time series data, our method can not make reasonable estimates without valid previous data. Therefore, the tracking performance may decrease significantly when packet losses sequentially occur for a long period.

To test randomly missing data, MSEs of each sample which are calculated by (6.23) and (6.24) are used instead of prediction MSEs in Kalman filter. Here we assume that every packet has the same probability to be lost (i.e., the network packet loss rate) and every event is independent. The simulation was continued for 100 synchronization periods and averaged over 200 runs when the sampling period is 900 seconds and the standard deviation of observation noise of clock offset is set to $3 \cdot 10^{-4}(s)$. We compare tracking performance of two methods. The first method used here is a "with tracking" method which is applied in the previous two missing cases. This method estimates clock skew and offset using the clock models and tracks them if no available observation. The other method is a "without tracking" method. Since this method does not process anything when observation does not exist, the estimates are not updated as a result and follow the previous estimates. By comparing with "without tracking" method, it is proved that our "with tracking" method is powerful in unstable networks. In Fig. 6.11, we plot the skew MSEs with different packet loss rates (10%, 20%, and 50%) using a "with tracking" method which tracks the clock skew and offset as finding their estimates with an AR process when the current data packet is not available. In the same figure, we also plot the "without tracking" cases which mean that the estimate keeps the same value as the previous sample when data missing. Both methods have performance degradation compared to MSE curves when no-missing. However, the "with tracking" method is robust to missing data compared with the "without tracking" method. The "without tracking" method has a significant performance degradation as depicted in Fig. 6.11. When loss rate is 20%, both MSEs of the "without tracking" method are higher as 10^4 times than the "with tracking" method. Even if we use the "without tracking" method, Kalman filter may not converge when 50% data are missing and thus the performance degrades. What we observe are: (i) as packet loss rate increases, the tracking performance gets worse, but our tracking method is fairly robust to

missing data; and (ii) performance with tracking always outperforms the one without tracking.

6.6.2 Tracking with Dirty Data

In networks, anomaly may happen everywhere any time. In this case, received data can be corrupted by other data packets, channel noise, or jamming. Our tracking method facilitates detecting corrupted data, combined with LASSO.

The corrupted data are called dirty data here and the model of the clock offset is redefined as:

$$\theta[n] = \tilde{\theta}[n] + v[n] + \xi[n], \tag{6.25}$$

where $v[n]$ is the Gaussian observation noise which usually has small variance and $\xi[n]$ is dirty component which has very large value once it happens.

- **Threshold based method**

Dirty data can be detected by a threshold based method and the removal of dirty data processes as follows:

(i) compute measurement noise; $\hat{v}[n] = \theta[n] - \hat{\theta}[n]$

(ii) compute $\frac{|\hat{v}[n]|}{\sigma_v}$ and compare with a pre-defined threshold; we determine it as dirty data when the value is greater than the threshold.

(iii) set $\theta[n] = \hat{\theta}[n]$ when the data is dirty.

(iv) repeat every synchronization period.

This method is intuitive, however, naive to detect dirty data.

- **LASSO (Least absolute shrinkage and selection operator)**

  Dirty data are seldom happened. Hence it can be solved as a constrained L_0 norm minimization problem [87].

$$minimize \quad \|\xi[n]\|_0 \quad subject \ \ to \ \ \theta[n] - \hat{\theta}[n] - v[n] = \xi[n], \tag{6.26}$$

  where $\|\cdot\|_0$ is the L_0 norm. However, since L_0 is not convex and is difficult to minimize, an approach to approximate the L_0 norm can be to replace the L_0 norm with an L_1 norm.

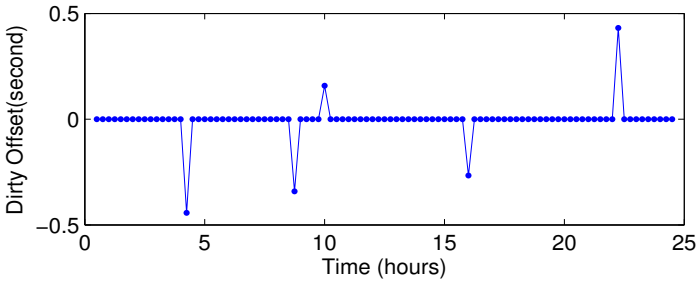
$$minimize \quad \|\xi[n]\|_1 \quad subject \ \ to \ \ \theta[n] - \hat{\theta}[n] - v[n] = \xi[n], \tag{6.27}$$

  where $\|\cdot\|_1$ is the L_1 norm. (6.27) is equivalent to

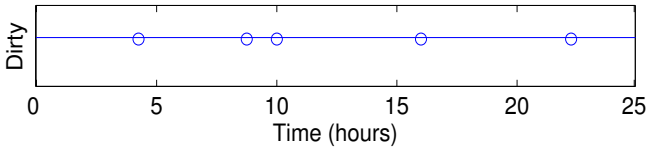
$$minimize \ \ \gamma \|\xi[n]\|_1 + \left\|\theta[n] - \hat{\theta}[n] - v[n] - \xi[n]\right\|_1, \tag{6.28}$$

  where $\gamma \in [0, 1]$ represents the weight of the constraints $\theta[n] - \hat{\theta}[n] - v[n] = \xi[n]$.

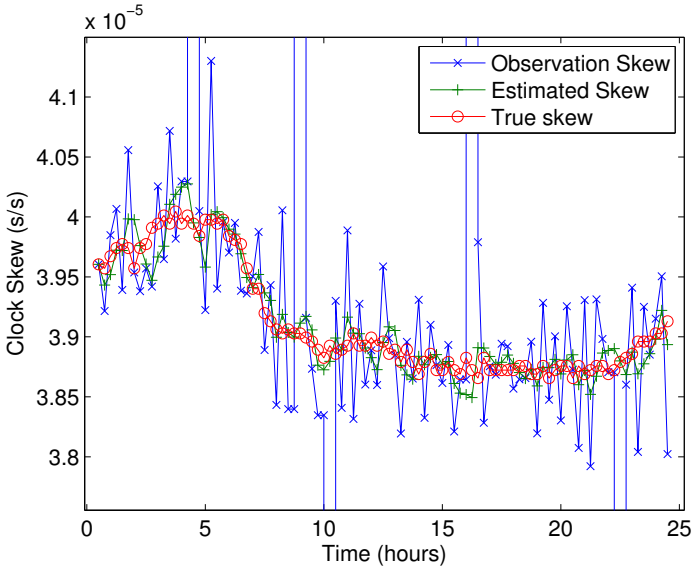
  To solve (6.27), we can apply LASSO method by adding a regularization term to a standard least squares problem [88]. LASSO minimizes the residual error with a constraint term of L_1 norm. Assuming $\psi[n]$ represents measurement noise and dirty part, i.e. $\psi[n] = v[n] + \xi[n]$, this problem



(a) Dirty Component (ξ)



(b) Estimated Dirty Data Position



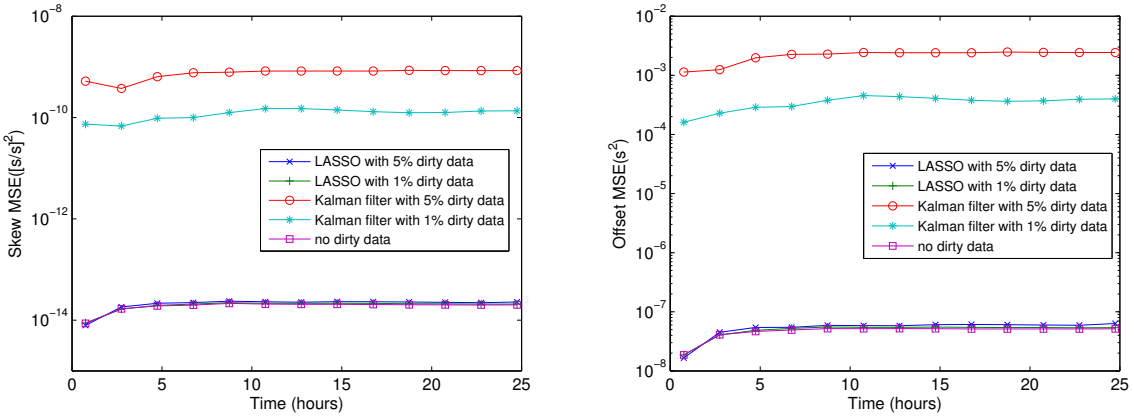
(c) Tracking of Clock Skew

Figure 6.12: Tracking with Dirty Data

becomes:

$$\hat{\psi}[n] = arg\min_{\psi[n]} \left\| \hat{\theta}[n] + \psi[n] - \tilde{\theta}[n] \right\|_2^2 + \lambda \left\| \psi[n] \right\|_1, \tag{6.29}$$

where $\|\cdot\|_2$ is the L_2 norm, and $\lambda \in [0, \infty)$. Here, since every term has only one component, this problem goes down to solve a quadratic equation of $\psi[n]$. By differentiating the quadratic equation, we can find an estimate $\hat{\psi}[n]$ to satisfy (6.29). If the estimate $\hat{\psi}[n]$ is larger than three standard deviations of measurement noise $3\sigma_v$ which accounts for 99.7%, this method determines that this observation has dirty part, and regards the observation as dirty data. As removing detected dirty data, this method avoids estimating clock behavior mistakenly. In this case, clock behaviors are estimated from AR process instead of observation data, which is dirty, as if data are missing. Since applying this process every samples, it is possible to decide if incoming sample is corrupted or not



(a) Skew MSE with Dirty Data

(b) Offset MSE with Dirty Data

Figure 6.13: MSEs with Dirty Data

and suitable for real-time synchronization.

Once a data is claimed as dirty, it is removed and treated as missing data. We assume the dirty component is randomly injected. The simulation was continued for 100 synchronization periods and averaged over 200 runs when a synchronization period is $900(s)$ and σ_v is $3 \cdot 10^{-4}(s)$. In the case that dirty data are generated as in Fig. 6.12a, our method using LASSO detects the position of dirty data exactly shown in Fig. 6.12b. We set λ as 0.01 for simulation. Our method discards these dirty data and tracks the clock skews and offset as replacing them by estimates based on the time-varying clock models as explained in Section 6.6.1. Fig. 6.12c represents that the estimates follow the true clock skew when observations have extremely large positive or negative values which are placed outside of this figure.

Fig. 6.13 presents the tracking performance when every data has 5% or 1% probability to be dirty data. The above two curves are the MSEs when dirty observations are not detected and the next two curves are those when dirty observations are detected and removed using LASSO, and the last curve represents the MSEs when dirty observation does not exist as a reference. Without removal of dirty data, the performance becomes much worse since the Kalman filter uses dirty data, which are incorrect clock information as observation. However, our approach prevents from deceasing the performance of tracking clocks by applying LASSO when the system has dirty data compared to the case when there is no dirty data. This example shows that our tracking method to utilize LASSO becomes robust to dirty data.

6.7 Clock Skew Estimation without Exchanging Timestamps

6.7.1 Main results

Suppose the are two nodes in one-hop distance. One of the nodes becomes a reference node, and the other node estimates the relative skew to the reference node's clock. We call the reference node as "node A" and a clock in the reference node as "clock A", and the other node as "node B" and its

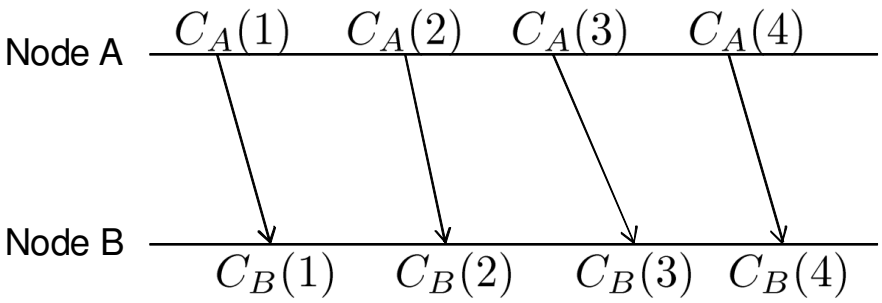


Figure 6.14: One-way timestamp transmissions

clock as "clock B". We suppose that both nodes share a common synchronization period ΔT using their own clock measures. Clock A and clock B are assumed to have the same initial time, in other words, they have the same offset at the beginning. Nodes can be set to have the same initial time before spreading out nodes to the field. If nodes do not have the same initial time, the reference node can transmit a timestamp once when starting clock synchronization process so that they can acquire the same initial time.

All clocks behave differently even if they have the same initial offsets, since all oscillators are random. The clock at each node is modeled as:

$$C_i(t) = \alpha_i t + \theta_i, \tag{6.30}$$

where i means A or B, θ_i is the clock offset, and α_i is the skew of the ith clock relative to the absolute time. We define a relative skew as a ratio of a node's skew to the other node's skew ($\alpha_{AB} = \frac{\alpha_A}{\alpha_B}$) [2]. The goal here is to estimate α_{AB} or α_{BA} through information exchange between two nodes.

Traditional skew estimation with timestamp exchanges

Here we calculate a relative skew using timestamps which have been widely adopted in [54, 55, 64–66]. For one-way synchronization methods, a reference node (node A) broadcasts timestamps periodically or aperiodically to announce its time as shown in Figure 6.14. Node B receives node A's timestamps after propagation delay and estimates the relative skew as:

$$\alpha_{BA}(i) = \frac{C_A(i) - C_A(i-1)}{C_B(i) - C_B(i-1)}. \tag{6.31}$$

Due to random propagation delay, the estimate of a relative skew can not be accurate. Moreover, since various factors make the oscillator nonlinear, the relative skew may be time-varying [79]. However, it changes relatively slow and in most cases, it is treated as a fixed parameter. When the propagation delay and/or the observation noise exist, least-squares or maximum-likelihood estimators can be derived to enhance the performance [89]. The main challenge for the estimator in (6.31) is that $C_A(i)$'s are real numbers which cost bandwidth and power to transmit from node A to node B. The quantified version of $C_A(i)$ degrades the performance.

[2] Different from some other chapters [60, 79] where a relative skew is defined as $\alpha_{AB} = \alpha_A - \alpha_B$

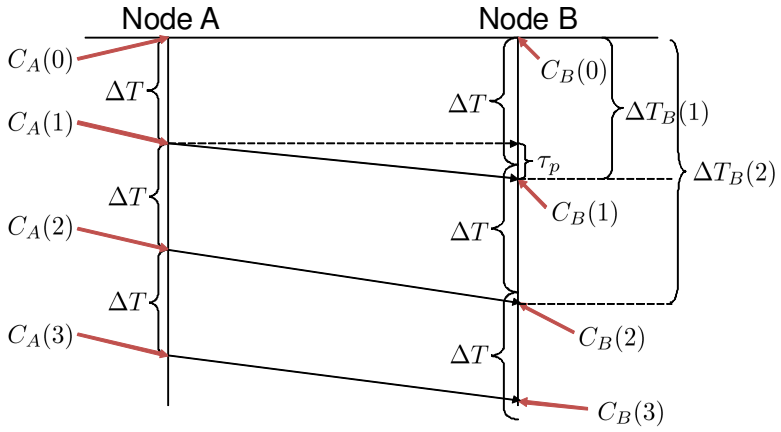


Figure 6.15: One bit transmissions

Proposed skew estimation

Figure 6.15 shows the proposed transmission protocol. Node A periodically sends node B a one bit signal, which can be embedded at the headers of multi-packet transmissions. Nodes A and B share a common notion of ΔT, which can be, e.g., a packet duration. Node A monitors its own clock and sends one bit to node B every ΔT seconds, and records the time instances. That does not mean that the absolute time t has passed ΔT. Based on clock model in (6.30), we have

$$\Delta T = C_A(i) - C_A(i-1) = \alpha_A \Delta t, \tag{6.32}$$

where α_A is the skew of clock A, and Δt is the absolute time difference. Node B receives one bit after propagation delay and records the receiving time instance $C_B(i)$ using its own clock. Again, based on the clock model in (6.30), at node B, we have

$$\Delta T_B(i) = C_B(i) - C_B(i-1) = \alpha_B \Delta t + \tau_p, \tag{6.33}$$

where α_B is the skew of clock B, and τ_p is random propagation delay difference. Since ΔT is known at both nodes but Δt is unknown, plugging (6.32) into (6.33) we have

$$\Delta T_B(i) = \alpha_B \frac{\Delta T}{\alpha_A} + \tau_p = \alpha_{BA} \Delta T + \tau_p. \tag{6.34}$$

Node B estimates the relative skew between node A and node B when receiving one bit as:

$$\hat{\alpha}_{BA} = \frac{\Delta T_B(i)}{\Delta T}. \tag{6.35}$$

When multiple one-bit signals are transmitted, multiple time instances have been recorded at node B. Suppose α is static and τ_p is zero-mean Gaussian distributed. The maximum-likelihood estimator for α_{BA} is

$$\hat{\alpha}_{BA} = \frac{1}{N\Delta T} \sum_{i=1}^{N} [C_B(i) - C_B(i-1)] = \frac{1}{N\Delta T} (C_B(N) - C_B(0)), \tag{6.36}$$

where N is the total number of observations. Note that to derive (6.36), we assume node A measures ΔT with no error in (6.32). However, this may not be practical. When noise exists in (6.32), we have

$$\Delta T = C_A(i) - C_A(i-1) = \alpha_A \Delta t + \eta_A, \tag{6.37}$$

where η_A is the observation noise at node A. Plugging (6.37) into (6.33), we have

$$\Delta T_B(i) = \frac{\alpha_B}{\alpha_A}(\Delta T - \eta_A) + \tau_p = \alpha_{BA}\Delta T - \alpha_{BA}\eta_A + \tau_p. \tag{6.38}$$

Suppose both τ_p and η_A are independent Gaussian distributed with zero-mean and variance σ_τ^2 and σ_η^2, respectively. The equivalent noise $n_{BA} = -\alpha_{BA}\eta_A + \tau_p$ is also Gaussian distributed with zero mean and variance $\sigma_n^2 = \sigma_\tau^2 + \alpha_{BA}^2\sigma_\eta^2$. Note that the noise variance depends on the unknown parameter α_{BA}. For simplicity, we consider two observations at node B and drop the index i. Since n_{BA} is Gaussian, the log-likelihood function based on (6.38) is

$$L(\alpha_{BA}) = -\frac{1}{2}ln(\sigma_\tau^2 + \alpha_{BA}^2\sigma_\eta^2) - \frac{1}{2\sigma_n^2}(\Delta T_B - \alpha_{BA}\Delta T)^2, \tag{6.39}$$

where we drop some constant terms which do not depend on α_{BA}. By setting the derivative of $L(\alpha_{BA})$ over α_{BA} to be zero, we obtain

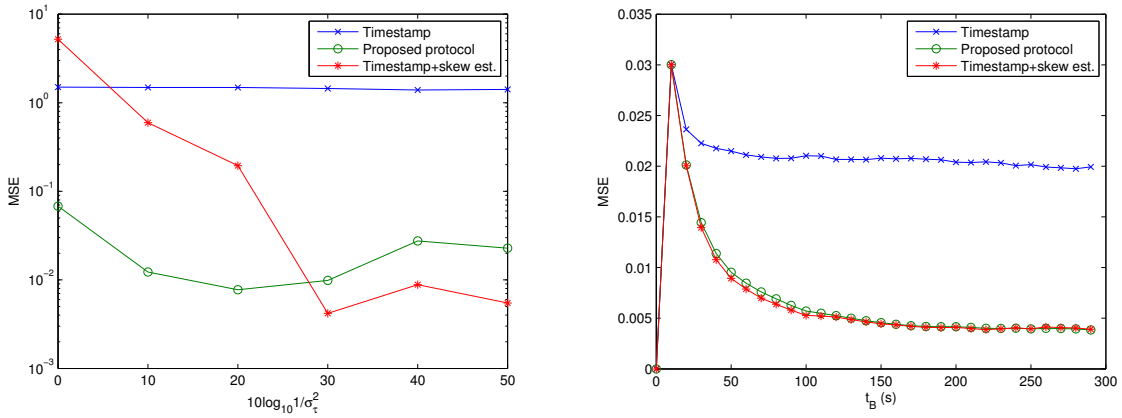
$$\sigma_n^2\sigma_\eta^2\alpha_{BA} + (\alpha_{BA}\Delta T - \Delta T_B)\sigma_n^2 - \alpha_{BA}\sigma_\eta^2(\Delta T_B - \alpha_{BA}\Delta T)^2 = 0. \tag{6.40}$$

Note that (6.40) is a third-order polynomial of α_{BA} and the root (i.e., the estimate of α_{BA}) can be found in closed form from [90]. This protocol can be extended to multiple nodes, more than two nodes in hierarchical multi-hop networks. Clocks of all nodes in a network finally will converge to the reference node's clock. The detailed performance analysis including Cramer-Rao-Bound (CRB) will be included in the full chapter.

6.7.2 Numerical Results

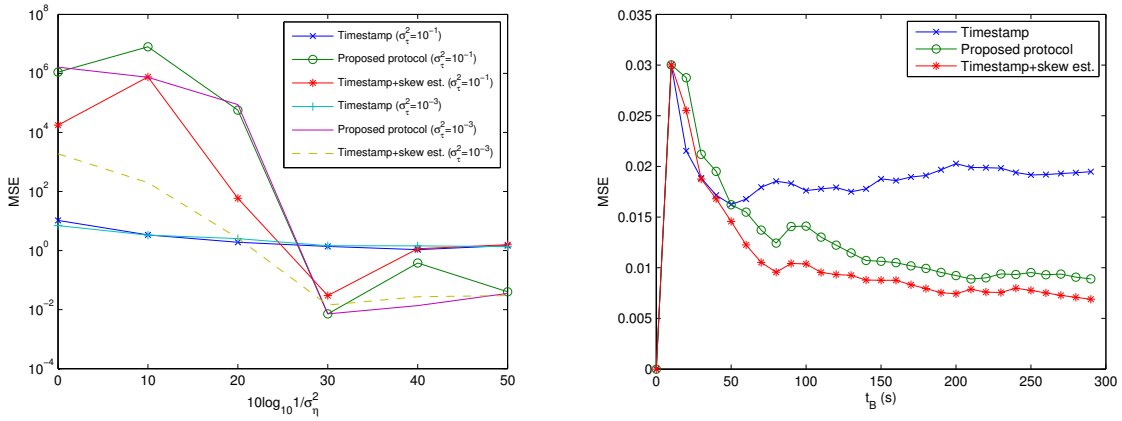
All oscillators have error. They may have a little bit different frequency from the originally intended when they are manufactured. Furthermore, the oscillating frequency is influenced by its age, supply voltage, power, and lots of environmental factors. This frequency error causes clock drift. Since an oscillator always behave erratically, clock skew is not 1 as mentioned in [57, 91]. We assume that nodes A and B flow at different speeds ($\alpha_A = 0.97, \alpha_B = 1.00$). Along the proposed protocol, node A transmits a one bit signal every synchronization period (ΔT=10). Node B receives one bit after propagation delay and estimates a relative skew α_{BA}.

- **case 1 (no observation noise at node A)**: Figure 6.16 compares three MSEs of popularly used timestamping method, the proposed protocol in 6.7.1 and timestamp based skew estimation in 6.7.1. Simulations are iterated as 50 times, and the propagation delay difference τ_p is randomly generated. Figure 6.16a shows MSEs regarding σ_τ^2 when $\Delta T = 10$ seconds, and N=2. The proposed protocol brings better performance than timestamping method when σ_τ^2 is less than 1. Figure 6.16b



(a) MSE vs. noise variance (b) MSE vs. time

Figure 6.16: MSEs without observation noise at node A



(a) MSE vs. noise variance (b) MSE vs. time

Figure 6.17: MSEs with observation noise at node A

illustrates the performance of time when $\Delta T = 10$ seconds and σ_τ^2 is 10^{-3}. The three MSE curves converge as time goes on. The proposed protocol has higher error than timestamp based skew estimation method, however, it has smaller error than timestamping method.

- **case 2 (observation noise at node A)**: Figure 6.17 represents the performance considering that the observation noise exists at node A. Simulations are iterated as 100 times. The propagation delay difference τ_p and observation noise at node A η_A are randomly generated with zero mean. Figure 6.17a shows MSEs regarding observation noise variance at node A when $\sigma_\tau^2 = 10^{-3}$, $\Delta T = 10$ seconds, and N=2. Only with two transmissions, the proposed protocol performs better than timestamping method when σ_η^2 is less than 10^{-3} in both case of $\sigma_\tau^2 = 10^{-1}$ and $\sigma_\tau^2 = 10^{-3}$. Figure 6.17b depicts MSEs when $\sigma_\tau^2 = 10^{-3}$ and $\sigma_\eta^2 = 10^{-3}$. All these three methods have higher MSE than those in Figure 6.16b due to the observation noise at node A.

In MSE comparisons, the timestamp based skew estimation method performs better than the proposed protocol. However, the proposed protocol consumes less transmission energy and bandwidth,

and lower complexity, since it uses only one bit signal for synchronization. These simulations show that the proposed protocol is comparable to timestamping method.

6.8 Conclusion

Efficient and accurate network time synchronization is a challenging problem in resource-constrained networks. Considering the inherent instability of the inexpensive oscillators, we show to derive the general models to capture the time-varying behavior of clock offset and skew. Then, applying these models we explain clock skew and offset tracking method based on the Kalman filter. This tracking method is analyzed in detail and evaluated with measurement of real low-cost clock behaviors. We demonstrate that our tracking method is robust to unreliable observations. Lastly, we develop an energy-efficient clock synchronization protocol without exchanging timestamps. We show that this technique performs better than timestamping method through simulations.

Bibliography

[1] S. Barbarossa, M. Pompili, and G. Giannakis, "Channel-independent synchronization of orthogonal frequency division multiple access systems," *IEEE J. Sel. Areas Commun.*, vol. 20, no. 2, pp. 474–486, Feb. 2002.

[2] Z. Cao, U. Tureli, and Y.-D. Yao, "Low-complexity orthogonal spectral signal construction for generalized OFDMA uplink with frequency synchronization errors," *IEEE Trans. Veh. Technol.*, vol. 56, no. 3, pp. 1143–1154, May 2007.

[3] S. Lee, H. Yoo, M. Park, B. Park, and D. Hong, "Timing synchronization using phase difference between subcarriers for OFDMA uplink systems over frequency selective fading channels," in *Proc. IEEE VTC 2005-Spring*, Stockholm, Sweden, May 30–Jun. 1, 2005, pp. 1258–1262.

[4] X. Fu, Y. Li, and H. Minn, "A new ranging method for OFDMA systems," *IEEE Trans. Wireless Commun.*, vol. 6, no. 2, pp. 659–669, Jun. 2007.

[5] M. Morelli, L. Sanguinetti, and H. V. Poor, "A robust ranging scheme for OFDMA-based networks," *IEEE Trans. Commun.*, vol. 57, no. 8, pp. 2441–2452, Aug. 2009.

[6] L. Sanguinetti, M. Morelli, and H. V. Poor, "An ESPRIT-based approach for initial ranging in OFDMA systems," *IEEE Trans. Commun.*, vol. 57, no. 11, pp. 3225–3229, Nov. 2009.

[7] *Air Interface for Fixed and Mobile Broadband Wireless Access Systems Amendment 2: Physical and Medium Access Control Layers for Combined Fixed and Mobile Operation in Licensed Bands and Corrigendum 1*, IEEE Std. 802.16e-2005, 2006.

[8] J. Liu and X. Liu, "An eigenvector-based approach for multidimensional frequency estimation with improved identifiability," *IEEE Trans. Signal Process.*, vol. 54, no. 12, pp. 4543–4556, Dec. 2006.

[9] C. G. Khatri and C. R. Rao, "Solutions to some functional equations and their applications to characterization of probability distributions," *Sankhyā: The Indian Journal of Statistics, Series A*, vol. 30, no. 2, pp. pp. 167–180, 1968. [Online]. Available: http://www.jstor.org/stable/25049527

[10] S. Lee and X. Ma, "An improved synchronization scheme for OFDMA systems with initial ranging transmissions," in *Proc. ASILOMAR 2010*, Nov. 7–10, 2010.

[11] J. Liu, X. Liu, and X. Ma, "Multidimensional frequency estimation with finite snapshots in the presence of identical frequencies," *IEEE Trans. Signal Process.*, vol. 55, no. 11, pp. 5179–5194, 2007.

[12] Recommendation, "Guidelines for evaluation of radio transmission technologies for IMT-2000," ITU-R, Tech. Rep. Rec. ITU-R M.1225, 1997.

[13] H. Lee, J. S. Kwak, H. Cho, Y.-H. Kwon, S. Moon, X. Wang, Y.-S. Choi, H. Yoo, S.-H. Park, C.-W. Huang, P.-H. Kuo, Y.-T. Hsieh, P.-A. Ting, and Z. Yan-Xju, "Proposed AWD text on the ranging structures for non-synchronized AMSs," IEEE, Tech. Rep. IEEE C802.16m-09/1092r3, 2009.

[14] *DRAFT Amendment to IEEE Standard for Local and Metropolitan Area Networks - Part 16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems - Advanced Air Interface*, IEEE Std. P802.16m/D5, 2010.

[15] M. Kas, B. Yargicoglu, I. Korpeoglu, and E. Karasan, "A survey on scheduling in IEEE 802.16 mesh mode," *Commun. Surveys Tuts.*, vol. 12, no. 2, pp. 205–221, 2010.

[16] K.-D. Lee and V. C. M. Leung, "Fair allocation of subcarrier and power in an OFDMA wireless mesh network," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 11, pp. 2051–2060, Nov. 2006.

[17] *IEEE Standard for Local and metropolitan area networks Part 16: Air Interface for Broadband Wireless Access Systems*, IEEE Std. 802.16-2009 (Revision of IEEE Std 802.16-2004), May 2009.

[18] K. Raghunath and A. Chockalingam, "SIR analysis and interference cancellation in uplink OFDMA with large carrier frequency/timing offsets," *IEEE Trans. Wireless Commun.*, vol. 8, no. 5, pp. 2202–2208, May 2009.

[19] M. Park, K. Ko, B. Park, and D. Hong, "Effects of asynchronous MAI on average SEP performance of OFDMA uplink systems over frequency-selective Rayleigh fading channels," *IEEE Trans. Commun.*, vol. 58, no. 2, pp. 586–599, Feb. 2010.

[20] "IEEE Standard for Local and Metropolitan Area Networks Part 16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems Amendment 2: Physical and Medium Access Control Layers for Combined Fixed and Mobile Operation in Licensed Bands and Corrigendum 1," *IEEE Std 802.16e-2005 and IEEE Std 802.16-2004/Cor 1-2005 (Amendment and Corrigendum to IEEE Std 802.16-2004)*, pp. 1–822, Feb. 2006.

[21] X. Ma, C. Tepedelenlioğlu, G. Giannakis, and S. Barbarossa, "Non-data-aided carrier offset estimators for OFDM with null subcarriers: identifiability, algorithms, and performance," *IEEE J. Sel. Areas Commun.*, vol. 19, no. 12, pp. 2504–2515, Dec. 2001.

[22] F. Gao and A. Nallanathan, "Identifiability of data-aided carrier-frequency offset estimation over frequency selective channels," *IEEE Trans. Signal Process.*, vol. 54, no. 9, pp. 3653–3657, Sept. 2006.

[23] Y. Yao and G. Giannakis, "Blind carrier frequency offset estimation in SISO, MIMO, and multiuser OFDM systems," *IEEE Trans. Commun.*, vol. 53, no. 1, pp. 173–183, Feb. 2005.

[24] Z. Cao, U. Tureli, and Y. Yao, "Deterministic multiuser carrier-frequency offset estimation for interleaved OFDMA uplink," *IEEE Trans. Commun.*, vol. 52, no. 9, pp. 1585–1594, Sept. 2004.

[25] L. Sanguinetti and M. Morelli, "A low-complexity scheme for frequency estimation in uplink OFDMA systems," *IEEE Trans. Wireless Commun.*, vol. 9, no. 8, pp. 2430–2437, Aug. 2010.

[26] H. Wang and Q. Yin, "Multiuser carrier frequency offsets estimation for OFDMA uplink with generalized carrier assignment scheme," *IEEE Trans. Wireless Commun.*, vol. 8, no. 7, pp. 3347–3353, Jul. 2009.

[27] P. Sun and L. Zhang, "Low complexity pilot aided frequency synchronization for OFDMA uplink transmission," *IEEE Trans. Wireless Commun.*, vol. 8, no. 7, pp. 3758–3769, July 2009.

[28] D. Huang and K. Letaief, "An interference-cancellation scheme for carrier frequency offsets correction in OFDMA systems," *IEEE Trans. Commun.*, vol. 53, no. 7, pp. 1155–1165, July 2005.

[29] T. Yucek and H. Arslan, "Carrier frequency offset compensation with successive cancellation in uplink OFDMA systems," *IEEE Trans. Wireless Commun.*, vol. 6, no. 10, pp. 3546–3551, Oct. 2007.

[30] D. Marabissi, R. Fantacci, and S. Papini, "Robust multiuser interference cancellation for OFDM systems with frequency offset," *IEEE Trans. Wireless Commun.*, vol. 5, no. 11, pp. 3068–3076, Nov. 2006.

[31] H. Liu and U. Tureli, "A high-efficiency carrier estimator for OFDM communications," *IEEE Commun. Lett.*, vol. 2, no. 4, pp. 104–106, Apr. 1998.

[32] R. Larson and B. Edwards, *Calculus*, 9^{th} ed. Brooks/Cole Pub. Co., 2009.

[33] X. Ma, M.-K. Oh, G. Giannakis, and D.-J. Park, "Hopping pilots for estimation of frequency-offset and multiantenna channels in MIMO-OFDM," *IEEE Trans. Commun.*, vol. 53, no. 1, pp. 162–172, Jan. 2005.

[34] M. Ozdemir and H. Arslan, "Channel estimation for wireless OFDM systems," *IEEE Commun. Surveys Tuts.*, vol. 9, no. 2, pp. 18–48, 2007.

[35] W. R. Heinzelman, J. Kulik, and H. Balakrishnan, "Adaptive protocols for information dissemination in wireless sensor networks," in *Proc. ACM MobiCom*, Seattle, WA, US, Aug. 1999, pp. 174–185.

[36] J. N. Al-Karaki and A. E. Kamal, "Routing techniques in wireless sensor networks: a survey," *IEEE Trans. Wireless Commun.*, vol. 11, no. 6, pp. 6–28, Dec. 2004.

[37] C. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers," in *Proc. ACM SIGCOMM*, London, UK, Oct. 1994, pp. 234–244.

[38] D. Johnson and D. Maltz, "Dynamic source routing in ad-hoc wireless networks," in *Proc. ACM SIGCOMM*, Stanford, CA, US, Aug. 1996, pp. 153–181.

[39] C. E. Perkins and E. M. Royer, "Ad-hoc on-demand distance vector routing," in *Proc. IEEE WMCSA*, vol. 3, Aug. 1999, pp. 90–100.

[40] B. Karp and H. T. Kung, "GPSR: greedy perimeter stateless routing for wireless networks," in *Proc. ACM MobiCom*, Aug. 2000, pp. 243–254.

[41] S. M. Das, H. Pucha, and Y. C. Hu, "Microrouting: a scalable and robust communication paradigm for sparse ad hoc networks," in *Proc. IEEE IPDPS*, Apr. 2005.

[42] S. Acedanski, S. Deb, M. Médard, and R. Koetter, "How good is random linear coding based distributed networked storage?" in *Proc. IEEE NETCOD*, Apr. 2005.

[43] A. G. Dimakis, P. B. Godfrey, M. J. Wainwright, and K. Ramchandran, "Network coding for distributed storage systems," in *Proc. IEEE INFOCOM*, Anchorage, AK, US, May 2007, pp. 2000–2008.

[44] A. G. Dimakis, J. Wang, and K. Ramchandran, "Unequal growth codes: intermediate performance and unequal error protection for video streaming," in *Proc. IEEE MMSP*, Chania, Greece, Oct. 2007, pp. 107–110.

[45] A. G. Dimakis, V. Prabhakaran, and K. Ramchandran, "Ubiquitous access to distributed data in large-scale sensor networks through decentralized erasure codes," in *Proc. IEEE IPSN*, Apr. 2005, p. 15.

[46] A. Jiang, "Network coding for joint storage and transmission with minimum cost," in *Proc. IEEE ISIT*, Seattle, WA, US, Jul. 2006, pp. 1359–1363.

[47] A. Kamra, V. Misra, J. Feldman, and D. Rubenstein, "Growth codes: maximizing sensor network data persistence," in *Proc. ACM SIGCOMM*, Pisa, Italy, Sept. 2006, pp. 255–266.

[48] J. W. Byers, M. Luby, M. Mitzenmacher, and A. Rege, "A digital fountain approach to reliable distribution of bulk data," in *Proc. ACM SIGCOMM*, Vancouver, Canada, Sept. 1998, pp. 56–67.

[49] S. Karande, K. Misra, and H. Radha, "Natural growth codes: partial recovery under random network coding," in *Proc. IEEE CISS*, Mar. 2008, pp. 540–544.

[50] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Médard, and J. Crowcroft, "XORs in the air: practical wireless network coding," in *Proc. ACM SIGCOMM*, Pisa, Italy, Sept. 2006, pp. 243–254.

[51] Y. Xin, Z. Wang, and G. B. Giannakis, "Space-time diversity systems based on linear constellation precoding," *IEEE Trans. Wireless Commun.*, vol. 2, no. 2, pp. 294–309, Mar. 2003.

[52] X. Ma and G. B. Giannakis, "Complex-field coded MIMO systems: performance, rate, and trade-offs," *Wireless Comm. and Mobile Computing*, pp. 693–717, Nov. 2002.

[53] Z. Wang and G. B. Giannakis, "Complex-field coding for OFDM over fading wireless channels," *IEEE Trans. Inf. Theory*, vol. 49, no. 3, pp. 707–720, Mar. 2003.

[54] D. L. Mills, "Internet time synchronization: the network time protocol," *IEEE Trans. on Comm.*, vol. 39, no. 10, pp. 1482–1493, Oct. 1991.

[55] ——, "Improved algorithms for synchronizating computer network clocks," *IEEE/ACM Trans. Networking*, vol. 3, pp. 245–254, Jun. 1995.

[56] C. Liao, M. Martonosi, and D. W. Clark, "Experience with an adaptive globally-synchronizing clock algorithm," in *Proc. ACM Symposium on Parallelism Algorithms and Architectures*, 1999, pp. 106–114.

[57] D. Veitch, S. Babu, and A. Pàsztor, "Robust synchronization of software clocks across the internet," in *Proc. ACM SIGCOMM IMC*, Oct. 2004, pp. 219–232.

[58] A. Pàsztor and D. Veitch, "PC based precision timing without GPS," *ACM SIGMETRICS Performance Evaluation Review*, vol. 30, pp. 1–10, Jun. 2002.

[59] "IEEE std. 1588 - 2002 IEEE standard for a precision clock synchronization protocol for networked measurement and control systems," *IEEE Std 1588-2002*, pp. i–144, 2002.

[60] S. B. Moon, P. Skelly, and D. Towsley, "Estimation and removal of clock skew from network delay measurements," in *Proc. INFOCOM*, vol. 1, Mar. 1999, pp. 227–234.

[61] L. Zhang, Z. Liu, and C. H. Xia, "Clock synchronization algorithms for network measurements," in *Proc. INFOCOM*, vol. 1, 2002, pp. 160–169.

[62] G. Pottie and W. Kaiser, "Wireless integrated network sensors," *Communications of ACM*, vol. 43, no. 5, pp. 51–58, May 2000.

[63] J. R. Vig, "Introduction to quartz frequency standards," Army Research Laboratory, Tech. Rep. SLCET-TR-92-1 (Rev. 1), Oct. 1992. [Online]. Available: http://www.ieee-uffc.org/freqcontrol/quartz/vig/vigtoc.htm

[64] J. Elson, L. Girod, and D. Estrin, "Fine-grained network time synchronization using reference broadcasts," *SIGOPS Oper. Syst. Rev.*, vol. 36, no. SI, pp. 147–163, 2002.

[65] S. Ganeriwal, R. Kumar, and M. Srivastava, "Timing-sync protocol in sensor networks," in *Proc. ACM SENSYS*, Nov. 2003.

[66] M. Maròti, G. S. B. Kusy, and A. Lèdeczi, "The flooding time synchronization protocol," in *Proc. ACM SENSYS*, Nov. 2004, pp. 39–49.

[67] M. Mock, R. Frings, E. Nett, and S. Trikaliotis, "Continuous clock synchronization in wireless real-time applications," in *Proc. IEEE SRDS*, Oct. 2000, pp. 125–133.

[68] S. PalChaudhuri, A. Saha, and D. Johnson, "Adaptive clock synchronization in sensor networks," in *Proc. IEEE IPSN*, Apr. 2004.

[69] W. Su and I. Akyildiz, "Time-diffusion synchronization protocols for sensor networks," *IEEE/ACM Trans. on Networking*, 2005.

[70] K. Romer, "Time synchronization in ad hoc networks," in *Proc. ACM MobiHoc*, Oct. 2001.

[71] M. Sichitiu and C. Veerarittiphan, "Simple, accurate time synchronization for wireless sensor networks," in *Proc. IEEE WCNC*, 2003.

[72] Q. Li and D. Rus, "Global clock synchronization in sensor networks," in *Proc. INFOCOM*, Mar. 2004.

[73] D. Zhou and T. Lai, "A scalable and adaptive clock synchronization protocol for ieee 802.11-based multihop ad hoc networks," in *Proc. IEEE Mobile Adhoc and Sensor Systems*, Nov. 2005.

[74] K. Kim and B. Lee, "KALP: A Kalman filter-based adaptive clock method with low-pass prefiltering for packet networks use," *IEEE Trans. on Comm.*, vol. 48, no. 7, Jul. 2000.

[75] A. Bletsas, "Evaluation of Kalman filtering for network time keeping," *IEEE Trans. on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 52, no. 9, Sep. 2005.

[76] L. Auler and R. d'Amore, "Adaptive Kalman filter for time synchronization over packet-switched networks (an heuristic approach)," in *Proc. IEEE COMSWARE*, Jan. 2007.

[77] "Clock oscillator stability," cardinal Components Inc. Applications Brief No. A.N. 1006, www.cardinalxtal.com/docs/notes.

[78] B. Sundararaman, U. Buy, and A. Kshemkalyani, "Clock synchronization for wireless sensor networks: a survey," *Ad Hoc Networks*, vol. 3, no. 3, Feb. 2005.

[79] B. R. Hamilton, X. Ma, Q. Zhao, and J. Xu, "ACES: adaptive clock estimation and synchronization using Kalman filtering," in *Proc. ACM MobiCom*, 2008, pp. 152–162.

[80] R. H. Jones and F. Boadi-Boateng, "Unequally spaced longitudinal data with ar(1) serial correlation," *Biometrics*, vol. 47, no. 1, pp. 161–175, 1991.

[81] G. Kitagawa and W. Gersch, *Smoothness priors analysis of time series*, 1st ed. Springer, 1996.

[82] P. Stoica and Y. Selen, "Model-order selection: a review of information criterion rules," *IEEE Signal Processing Magazine*, vol. 21, pp. 36–47, Jul. 2004.

[83] A. Barron, J. Rissanen, and B. Yu, "The minimum description length principle in coding and modeling," *IEEE Signal Processing Magazine*, vol. 44, pp. 2743–2760, Oct. 1998.

[84] M. H. Hayes, *Statistical Digital Signal Processing and Modeling*, 1st ed. Wiley, 1996.

[85] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*, 1st ed. Prentice Hall, 1993.

[86] Crossbow, Inc., "Mica2 datasheet." [Online]. Available: http://www.xbo
w.com/Products/Product_pdf_files/Wireless_pdf/MICA2_Datasheet.pdf

[87] Y. Zhang, A. G. Z. Ge, and M. Roughan, "Network anomography," in *Proc. ACM SIGCOMM IMC*, Oct. 2005.

[88] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J. R. Statist. Soc. B*, vol. 58, no. 1, pp. 267–288, 1996.

[89] K. Noh, Q. M. Chaudhari, E. Serpedic, and B. W. Suter, "Novel clock phase offset and skew estimation using two-way timing message exchanges for wireless sensor networks," *IEEE Trans. on Comm*, vol. 55, no. 4, Apr. 2007.

[90] [Online]. Available: http://http://en.wikipedia.org/wiki/Cubic_function.

[91] H. Kim, X. Ma, and B. Hamilton, "Modeling and tracking clocks with time-varying skews using information criteria," in *Proc. MILCOM*, Oct. 2010.